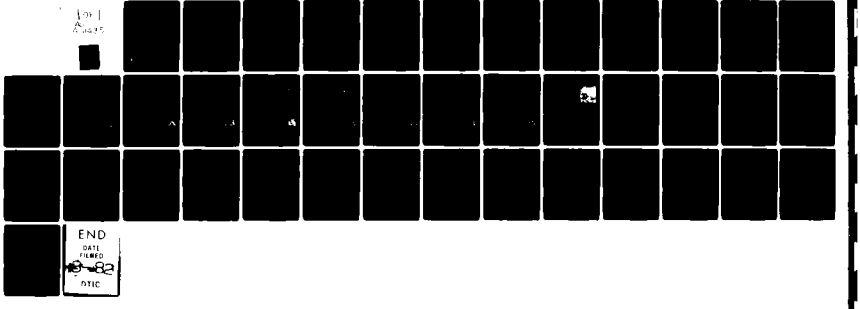
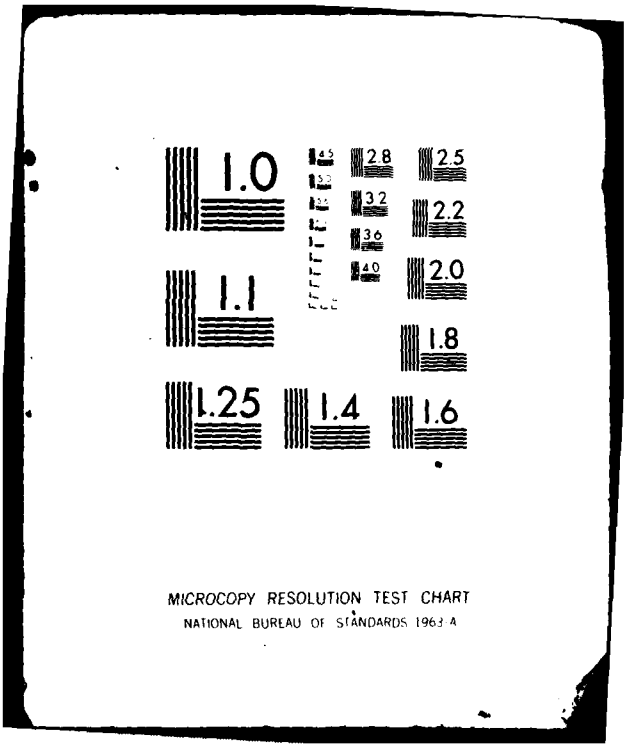
AD-A111 435 SOUTHEASTERN MASSACHUSETTS UNIV NORTH DARTMOUTH DEPT --ETC F/G 9/2
A FURTHER STUDY OF THE MULTICHANNEL MAXIMUM ENTROPY SPECTRAL AN--ETC(U)
FEB 82 C H CHEN, G YOUNG N00014-79-C-0494
UNCLASSIFIED SMU-EE-TR-82-03 NL



END
DATE FILED
4-82
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Technical Report
SMU-EE-TR-82-02
Contract Number N00014-79-C-0494
February 19, 1982

(1)

ADA111435

A Further Study of the Multichannel Maximum Entropy Spectral Analysis*

By
C. H. Chen
Gia-Kuen Young
Department of Electrical Engineering
Southeastern Massachusetts University
North Dartmouth, Massachusetts 02747

*The support of the Statistics and Probability Program of the Office of Naval Research on this work is gratefully acknowledged.

A Further Study of the Multichannel Maximum Entropy Spectral Analysis

I. Introduction

In the field of signal detection and estimation, it is always necessary to face the problems with systems whose states are described by several time variables. For example, in the geophysical research, data are gathered by arrays of sensors (seismographs, hydrophones, antennas, meteorological stations). The sensors used in these array systems will not always be along a line or in a regular grid but will often have some arbitrary spatial distribution. Also, in an array system, one or more sensors can be connected to a channel; each of these channels provides one of the signal to be processed. Instead of processing the signals received on each channel separately, it is possible to process all the received signals collectively by means of a multichannel filter that takes into account the interrelations among the individual signals. Therefore, multichannel time series processing is quite important because of its sufficient informations about the object. To solve this problem, spectral estimation is an effective tool to describe the signal properties. Various attempts [1] - [4], [8] - [9] of multichannel spectral analysis have been made in the literature. Many conventional methods have been extended from one channel to multichannel. One technique which has been studied extensively due to its high resolution characteristics is the maximum entropy method (MEM). Jones [3], Nuttal [4], Strand [1] and Morf et al. [2] have independently obtained the multichannel generalization of the scalar Burg's maximum entropy method. In this report, we use the generalization method of Strand

as a means of study. Detailed computer results are presented for the multichannel maximum entropy spectral analysis of short sinusoids. Although for the real signals the spectral line shifting and spontaneous line splitting are observed, we have demonstrated that such phenomena are almost completely eliminated in the complex signal case. This result is practically significant as it indicates that accurate spectral estimation is possible for the complex signal case. The multichannel maximum entropy spectral analysis can thus provide a good phase estimate from which time delays among spatially separated sensors can be properly determined.

This report is organized as follows. In section II, we briefly review the algorithm of Strand. In section III, we show the results of the spectral estimation and the phase difference between channels. In section IV, we present the comparative result of the real signals and the complex signals in the viewpoint of line shifting and splitting properties. And in the Appendix, we list a detailed computer program in Fortran which we used to obtain the complex multichannel maximum entropy spectral estimation.

II. Theoretical Background

Let

$$y(t)=\begin{vmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_p(t) \end{vmatrix} , \qquad t=\{n\Delta t\}_{n=-\infty}^{\infty} \tag{1}$$

be a complex zero-mean, wide-sense stationary p-channel time series. The N-element forward filter (or forward filter of length N)

has the form.

$$y(t) + \sum_{k=1}^{N} F^{*}_{kN}\, y(t-k\Delta t) = e_N(t) \qquad (2)$$

Similarly, we consider the N-element backward filter

$$y(t) + \sum_{k=1}^{N} B^{*}_{kN} y(t+k\Delta t) = b_N(t) \qquad (3)$$

In (2) and (3), the forward and backward filter coefficients, F_{kN} and B_{kN}, respectively, are complex pxp matrices. The asterisk * denotes the (Hermitian) conjugate transpose. For the optimum forward or backward filters (2) or (3), we impose the condition that the expected mean-square value of $e_N(t)$ or $b_N(t)$ should be a minimum. For (2) this involves determining F_{1N}, F_{2N}, ... , F_{NN} such that

$$E\left\{ [e_N(t)]^{*} [e_N(t)] \right\} = \text{minimum} \qquad (4)$$

where $E\{\ \}$ denotes the expected value. Inserting (2) into (4) and minimizing with respect to the coefficients F_{kN}^{*} gives the system

$$\begin{bmatrix} R_0 & R_1 & R_2 & \cdots & R_N \\ R_{-1} & R_0 & R_1 & \cdots & R_{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ R_{-N} & R_{-N+1} & R_{-N+2} & & R_0 \end{bmatrix} \begin{bmatrix} I \\ F_{1N} \\ \vdots \\ F_{NN} \end{bmatrix} = \begin{bmatrix} P_N \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad (5)$$

where I is the pxp identity matrix and the square block submatrices in this system are defined by

$$R_k = E\left\{ y(t)\, y^{*}(t - k\Delta t) \right\} \quad , \quad k=0,1,\ldots N \qquad (6)$$

so that

$$R_{-k} = R_k^{*} \qquad (7)$$

We call the coefficient matrix in (5) the R-matrix, or simply R. The forward power matrix P_N for the resulting optimum filter (that is, for F_{kN} satisfying (5)) is

$$P_N = E\left\{e_N(t)\ e_N^*(t)\right\} \tag{8}$$

By multiplying both sides of (5) by F_N^* we have,

$$P_N = F_N^*\ R\ F_N \tag{9}$$

where we define F_N by

$$F_N^* = \left[I, F_{1N}^*, \ldots, F_{NN}^*\right]$$

with comma indicating matrix partition.

Similarly, for the optimum backward filter, we can derive a system:

$$\begin{bmatrix} R_0 & R_{-1} & R_{-2} \cdots & R_{-N} \\ R_1 & R_0 & R_{-1} \cdots & R_{-N+1} \\ \vdots & \vdots & \vdots & \vdots \\ R_N & R_{N-1} & R_{N-2} \cdots & R_0 \end{bmatrix} \begin{bmatrix} I \\ B_{1N} \\ \vdots \\ B_{NN} \end{bmatrix} = \begin{bmatrix} P_N' \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{10}$$

We call the coefficient matrix in (10) the R' matrix, or simply R'. The backward power matrix p_N' for the resulting optimum filter is

$$P_N' = E\left\{b_N(t)\ b_N^*(t)\right\} \tag{11}$$

and it follows from (10) that

$$P_N' = B_N^*\ R'\ B_N \tag{12}$$

where

$$B_N^* = \left[I,\ B_{1N}^*,\ \ldots,\ B_{NN}^*\right]$$

Burg has derived the following results concerning the recursive solution of (5) and (10), the forward and backward filters may be postulated in the form

$$F_N = \begin{bmatrix} F_{N-1} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ B'_{N-1} \end{bmatrix} C_N \qquad \text{and}$$

$$B_N = \begin{bmatrix} F_{N-1} \\ 0 \end{bmatrix} C'_N + \begin{bmatrix} 0 \\ B_{N-1} \end{bmatrix} \tag{13}$$

where $B'_{N-1} = \left[B^*_{N-1,N-1} \; \cdots \; B^*_{1,N-1} \; I \right]^*$ is the block reverse of B_{N-1}, and the pxp matrices C_N and C'_N are called forward and backward reflection coefficients, respectively. It can be shown that P_N and P'_N may be computed by using the modern Levinson algorithm:

$$P_N = P_{N-1} - C_N^* P'_{N-1} C_N \tag{14}$$

$$P'_N = P'_{N-1} - (C'_N)^* P_{N-1} C'_N \tag{15}$$

and

$$C_N = (P'_{N-1})^{-1} (C'_N)^* P_{N-1}$$

and

$$C'_N = (P_{N-1})^{-1} C_N^* P'_{N-1} \tag{16}$$

For the positive definite $P_N > 0$ (and $P_N' > 0$), it is known that the maximum entropy (autoregressive) power spectral density, S(f), a pxp matrix, can then be calculated from the forward filter F_N by

$$S(f) = \Delta t \left[F^{-1}\left(\frac{1}{z}\right) \right]^* P_N \left[F^{-1}\left(\frac{1}{z}\right) \right] \tag{17}$$

where $F(z) = I + F_{1N} z + \cdots + F_{NN} z^N$, z is the complex scalar defined by $z = \exp\left[-2\pi i f \Delta t\right]$ and f is the frequency.

It can be shown (as suggested in (9)) that $P_0, P_1, \ldots P_N$ and $(P_0', P_1', \ldots P_N')$ are positive definite if and only if R is positive definite. Instead of trying to solve (5) using approximations to the R_K, Strand [1] derived a method of estimating the reflection coefficients

C_N and C'_N by minimizing a certain weighted sum of squares of residuals for the forward and backward filters, after which all forward and backward filter coefficients $F_{k,N}$ and $B_{k,N}$ are obtained by the modern Levinson algorithm. It can be shown that

$$B\,C_N + P'_{N-1}\,C_N\,P'_{N-1}\,E = -2G \tag{18}$$

where

$$\begin{aligned} E &= \sum_{m=1}^{M} \bar{W}_m\, e_m^N\, (e_m^N)^* \\ G &= \sum_{m=1}^{M} \bar{W}_m\, b_m^N\, (e_m^N)^* \\ B &= \sum_{m=1}^{M} \bar{W}_m\, b_m^N\, (b_m^N)^* \end{aligned} \tag{19}$$

and $\bar{W}_m$ are positive scalar weights (usually all taken as $\frac{1}{M}$ in practice). Nd is the number of observation of a realization of $y(t)$, N is the number of tuples of data.

Also e_m^N and b_m^N can be obtained from

$$e_m^N = e_m^{N-1} + C_{N-1}^*\, b_{m+1}^{N-1}$$

and

$$b_m^N = b_m^{N-1} + (C'_{N-1})^*\, e_m^{N-1} \tag{20}$$

$$m=1,2,\ldots Nd-N = M\ ,\ N > 1$$

The whole computing procedure is illustrated in Fig. 1.

$P_0 = P_0' = \left(\sum_{m=1}^{Nd} y_m\, y_m^*\right)/Nd$

Let $e_m' = y_{m+1}$ and $b_m' = y_m$, $m=1,2,\ldots Nd-1$

↓

Compute E, B, G by (19)

↓

Solve C_N and C'_N by (18) and (16) respectively.

↓

Compute P_N and P'_N by (14) (15)

↓

Update F_N and B_N by (13)

↓

Obtain new e_m^N and b_m^N by (20)

↓

Calculate spectra by (17)

Fig. 1 Computing procedure for the multichannel Maximum Entropy method

III. Numerical Results of the Multichannel MEM Spectrum and Estimation of Phase Differences Between Channels

To ensure the correctness of the program, the same data which Strand used have been tested:

Nd (number of data points sampled) = 128

Δt = 1 second.

p (number of channels) = 2

$$y_t = \begin{bmatrix} y_{1t} \\ y_{2t} \end{bmatrix} \quad , \quad t=1,2,\ldots, \quad 128$$

$$\begin{aligned} \text{Re } y_{1t} &= \cos(2\pi f_i t) + 0.25\,(\text{ran f} - 0.5) \\ \text{Im } y_{1t} &= \sin(2\pi f_i t) + 0.25\,(\text{Ran f} - 0.5) \\ \text{Re } y_{2t} &= \cos(2\pi f_i t + \phi_d) + 0.25(\text{Ranf} - 0.5) \\ \text{Im } y_{2t} &= \sin(2\pi f_i t + \phi_d) + 0.25\,(\text{Ranf} - 0.5) \end{aligned} \tag{21}$$

where Ranf is a pseudo-random number uniformly distributed between 0 and 1, ϕ_d is the phase difference between channels and f_i is the input frequency. If we denote the spectrum by

$$S(f) = \begin{bmatrix} S_{11}(f) & S_{12}(f) \\ S_{21}(f) & S_{22}(f) \end{bmatrix} \tag{22}$$

where f is the frequency in Hz, then the squared coherence is given by

$$\text{coh}^2(f) = \frac{|S_{12}(f)|^2}{S_{11}(f)\,S_{22}(f)} \tag{23}$$

A rough indication of the expected performance is given by Akaike's FPE criterion, given by

$$\text{FPE}(N) = \det P_n \left(\frac{Nd + 1 + pN}{Nd - 1 - pN} \right)^p$$

Figs. 2a and 2b show $S_{11}(f)$ and $S_{22}(f)$ for the real sinusoid inputs with f= 0.0625 Hz, $\Delta\phi$d = 1 rad. and NPEF (number of terms in the prediction error filter) = 7. The results for $S_{12}(f)$ are shown (in amplitude and phase) in Figs. 2c and 2d. From Fig. 2d, we can observe that the phase of $S_{12}(f)$ appears to equal -1 radian at f=0.0625 Hz. In Fig. 2e, the squared coherence is shown. Fig. 2f and Fig. 2g also indicate the maximum and minimum eigenvalues of spectra. Figs. 3a-3g is the repeated procedure of Figs. 2a-2g but with complex input signals. The comparison shows that the complex multichannel MEM gives better resolution and accuracy than the real case.

It is well known that the phase difference between channels is a very important parameter. Correct phase difference can give very good estimation of time delay so as to detect the location of the target. Different sets of data have been tested using the complex multichannel MEM program with phase difference $\Delta\phi$d= 0.5 rad,-1.rad, 2. rad. (f_i= 0.0625 Hz), $\Delta\phi$d= -0.5 rad (f_i=0.2 Hz). The results of 1-1 spectrum $S_{11}(f)$ and phase of 1-2 spectrum, arg. $S_{12}(f)$ are shown in Figs. 4(a,b), 5(a,b), 6(a,b) and 7(a,b) respectively.

It is noted that the results are satisfactory.

IV. The Splitting and Shifting Problem in Multichannel MEM Power Spectra

In 1974, Chen and Stegen [7] ran some computer experiments. They noted that the estimated spectral maximum was found to shift in relation to the true maximum. The frequency shifts depended upon the initial phase and the length of the sinusoid. Fougere et al. [6] showed that the problem arose from the assumption of Burg that

previously determined reflection coefficient are held fixed in order to determine the present reflection coefficients. Also, Bougere [5] recently indicated that these properties occurred to the single channel complex signals and the multichannel real signals. In order to study this problem we have tested a signal which exemplified the problem for the single-channel signals but in both real and complex cases. This is a single sine wave of frequency 1 Hz sampled every 0.05 seconds, with additive noise:

$$\text{Re } f_1(n) = \cos((n-1)\pi /10 + \phi_1) + Ag_1(n)$$
$$\text{Im } f_1(n) = \sin((n-1)\pi /10 + \phi_1) + Ag_1(n)$$
$$\text{Re } f_2(n) = \cos((n-1)\pi /10 + \phi_2) + Ag_2(n)$$
$$\text{Im } f_2(n) = \sin((n-1)\pi /10 + \phi_2) + Ag_2(n)$$

where ϕ_1 and ϕ_2 are initial phases in each channel and $g_1(n)$ and $g_2(n)$ are independent realizations of random numbers in the range ± 0.5 and A is the noise amplitude. Also, we used Nd and NPEF to represent the number of data points and the number of terms in the prediction error filter respectively.

We have noted that the number of terms of filter (NPEF) affect the quality of the spectrum as indicated by Fig. 8a with NPEF=3,9,21. The same data are applied to the real and complex multichannel cases as shown in Fig. 8b, 8c respectively. We can see the real multichannel case is worse even than the single channel case (for NPEF=21), but the complex signal performs good. Figs. 9-11 show the effects of the number of data sample (Nd) to the location of spectral peak of real signals. Figs. 9a, 10a, 11a show the location of spectral peak is farther from 1 Hz as the cycles are not

completed. When the number of data points increases, the frequency shift decreases as shown in Fig. 14. As expected there is a node at Nd=21, 41, ..., every full cycle, where there is excellent agreement between true and apparent frequencies. However, Figs. 12a, 13a show two more cases but with different initial phase angles. The locations of spectral peak for various number of data are also shown in Figs. 15 and 16 respectively (for Ψ_1=45, Ψ_2=135 and Ψ_1=-45, Ψ_2=-135). Again we have noted the same property which is affected by the number of data points sampled. Comparing the location of the peak of Nd=21 of Figs. 9a, 12a and 13a, and the vertical axis of Figs. 14, 15, and 16, we have noted the peak of the real multichannel case is also affected by the initial phase. The peak with 45°, 135° phases shifts farther than with 0°, 180° case. This is the same condition as in the single channel case.

A very interesting property is that the location of spectral peaks does not shift away from the true value for various variations (number of data sampled, number of terms in the prediction error filter, value of initial phase). From Figs. 9b-13b and Figs. 14-16, the complex multichannel case always gives very satisfactory results. Therefore, we can conclude that the complex MEM power spectrum estimation is not affected by the factors indicated above because of its sufficient informations.

References

1. O. N. Strand, "Multichannel complex maximum entropy (autoregressive) spectral analysis," IEEE Trans. Automat. Contr., vol. AC-22, no. 4, pp. 634-640, August 1977.

2. M. Morf, A. Vieira, D. T. L. Lee and T. Kailath, "Recursive multichannel maximum entropy spectral estimation," IEEE Trans. on Geoscience Electronics, vol. GE-16, no. 2, pp. 85-94, April 1978.

3. R. H. Jones, "Multivariate autoregression estimation using residuals," in Applied Time Series Analysis edited by D. F. Findley, Academic Press, 1978.

4. A. H. Nuttall, "Fortran program for multivariate linear predictive spectral analysis employing forward and backward averaging," NUSC Technical Document 5419, Naval Underwater Systems Center, New London, Conn. 1976.

5. J. P. Fougere, "Spontaneous line splitting in multichannel maximum entropy power spectra," Proceedings of the 1st ASSP Workshop on Spectral Estimation, Hamilton, Ontario, Canada, Aug. 17-18, 1981. pp. 1 6 1 - 1 6 3.

6. J. P. Fougere, "A solution to the problem of spontaneous line splitting in maximum entropy power spectrum analysis," J. Geophys. vol. 82, no. 7, pp. 1051-1054, March 1977.

7. W. Y. Chen and G. R. Stegen, "Experiments with maximum entropy power spectra of sinusoids," J. Geophys. vol. 79, pp. 3019-3022, 1974.

8. E. A. Robinson, "Multichannel Time Series Analysis with Digital Computer Programs," Holden-Day, 1967.

9. R. A. Wiggins and E. A. Robinson, "Recursive solution to the multichannel filtering problem," J. Geophys. vol. 70, pp. 1885-1891, 1965.

10. C. H. Chen and Chisung Yen, "On multichannel (multivariate) maximum entropy spectral analysis," Technical Report, SMU-EE-TR-81-2, January 23, 1981.

11. J. Makhoul, "Linear prediction: a tutorial review," Proc. IEEE, vol. 63, no. 4, pp. 561-579, 1975.

12. T. J. Ulrych and T. N. Bishop, "Maximum entropy spectral analysis and autoregressive decomposition," Rev. Geophys. Space Phys., vol. 13, no. 1, pp. 183-200, 1975.

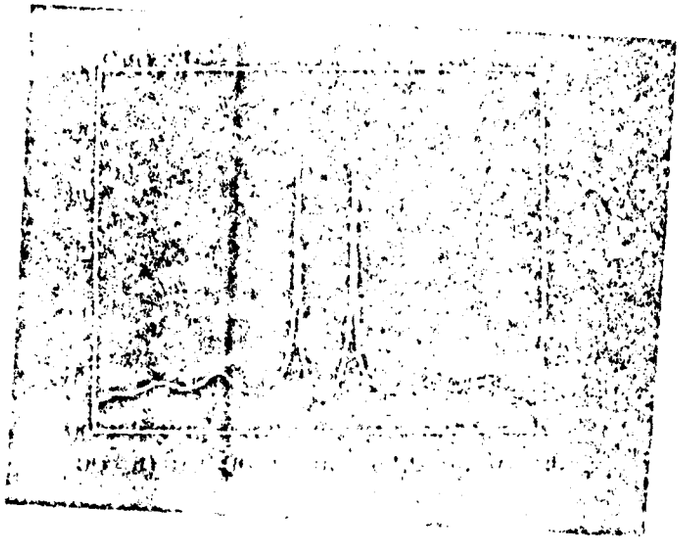


Fig. 2a 1-1 spectrum, $S_{11}(f)$, for real signal.

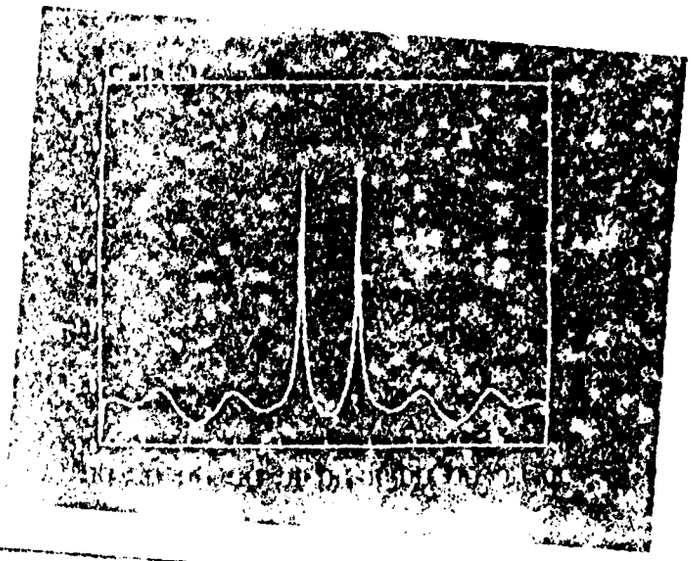


Fig. 2b 2-2 spectrum, $S_{22}(f)$, for real signal.

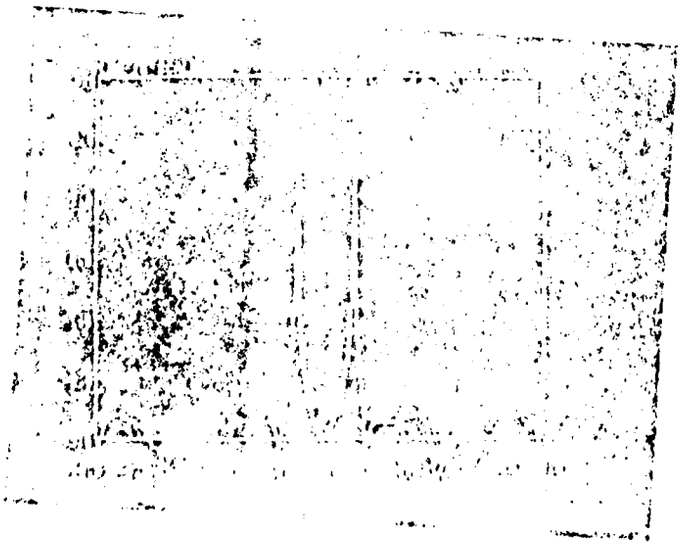


Fig. 2c Absolute value of 1-2 spectrum $|S_{12}(f)|$ for real signal.

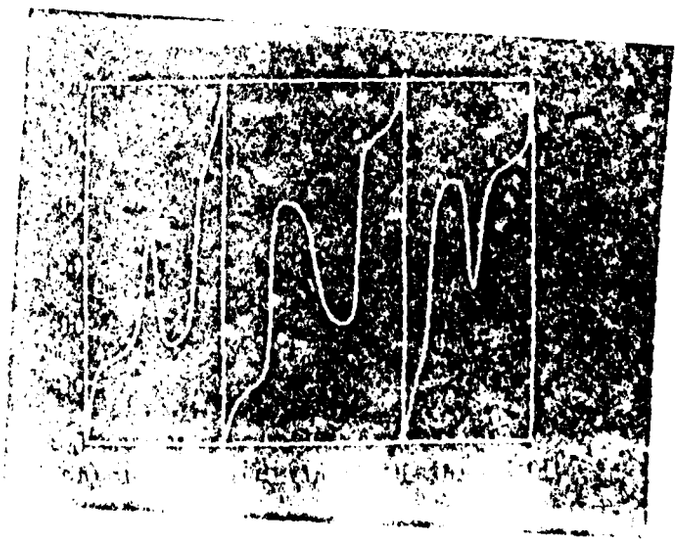


Fig. 2d Phase of 1-2 spectrum, $\arg S_{12}(f)$ for real signal.

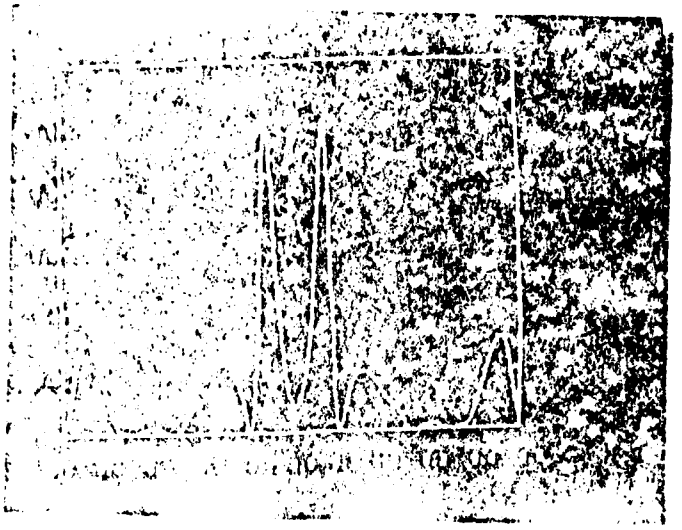


Fig. 2e [illegible] spectrum coherence squared, for real signal.

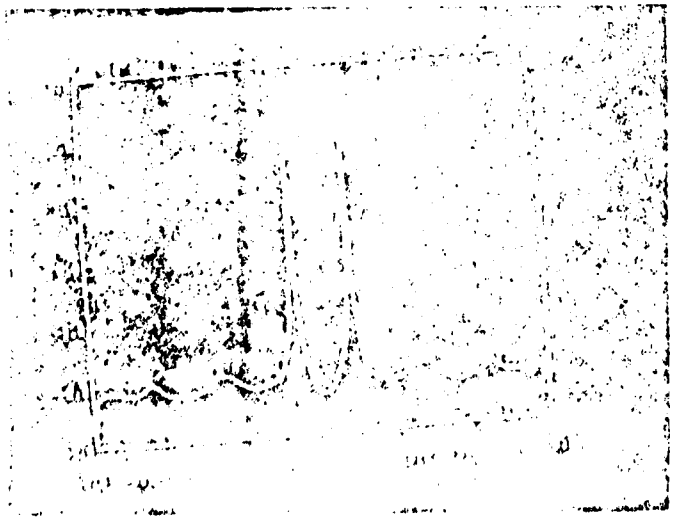


Fig. 2f Maximum eigenvalues of spectrum for real signal.

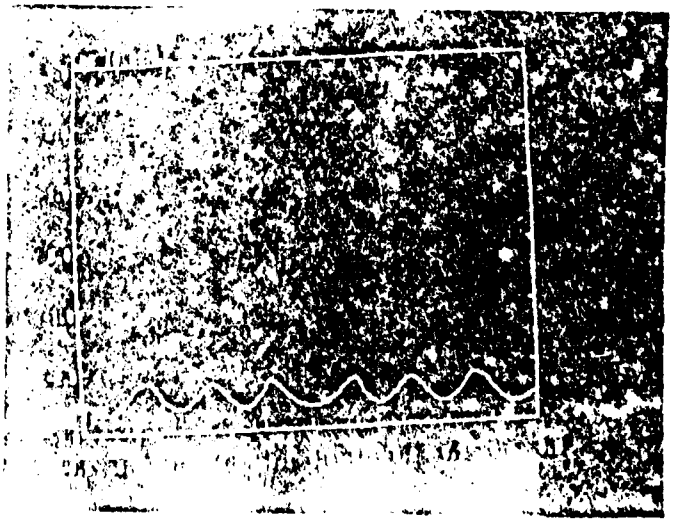


Fig. 2g Minimum eigenvalues of spectrum for real signal.

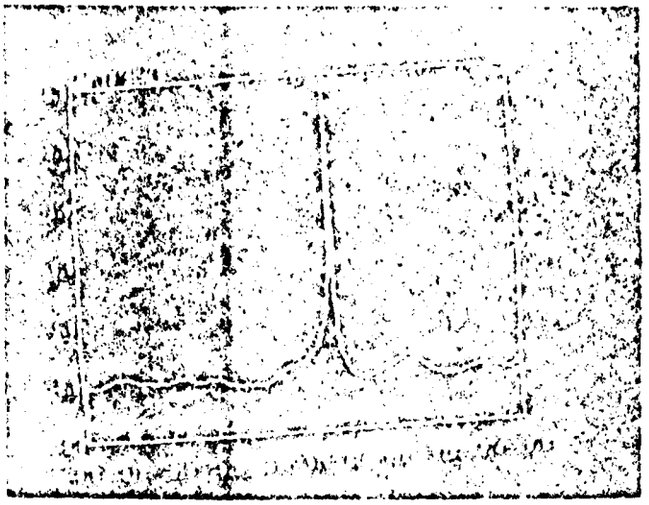


Fig. 3a 1-1 spectrum, $S_{11}(f)$ for complex signal.

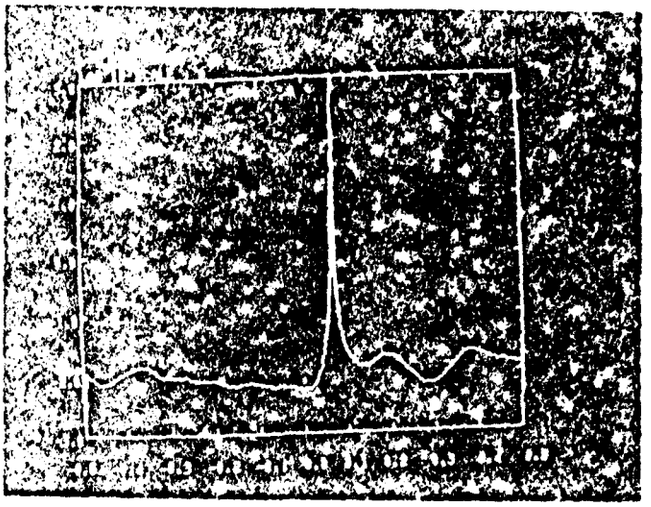


Fig. 3b 2-2 spectrum, $S_{22}(f)$ for complex signal.

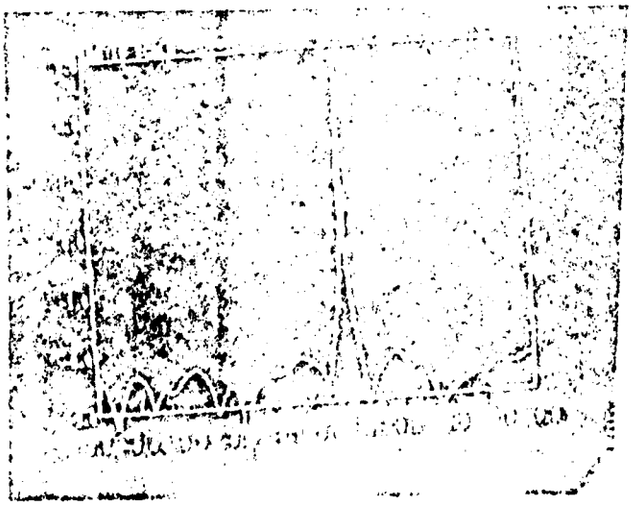


Fig. 3c Amplitude value of 1-2 spectrum $|S_{12}(f)|$ for complex signal.

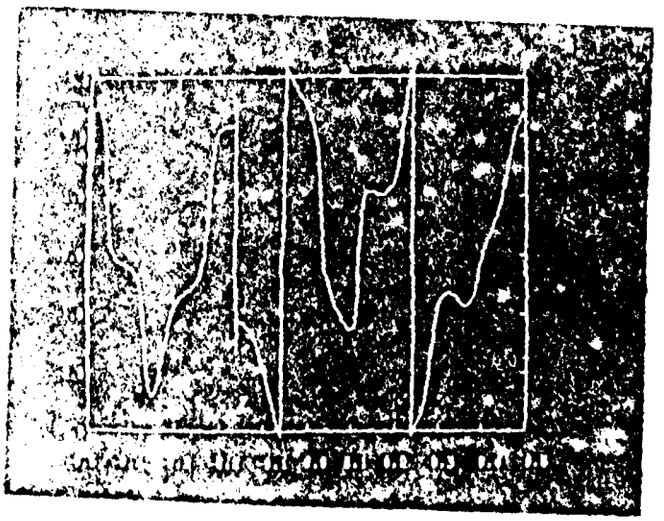


Fig. 3d Phase of 1-2 spectrum $\arg S_{12}(f)$ for complex signal.

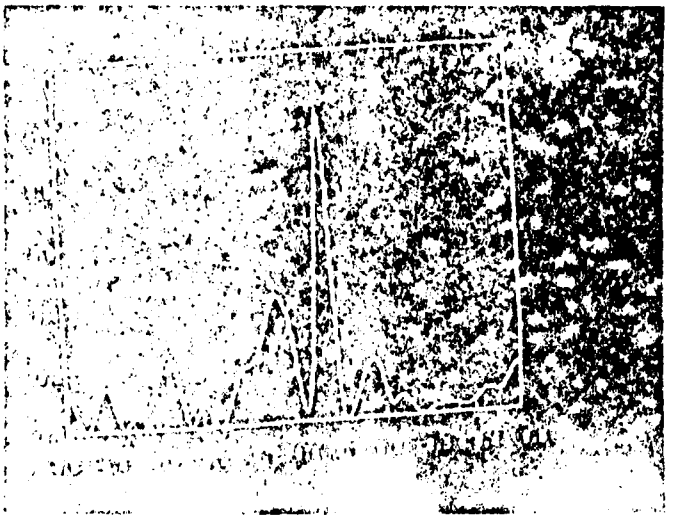


Fig. 3e Spectrum coherence squared for complex signal.

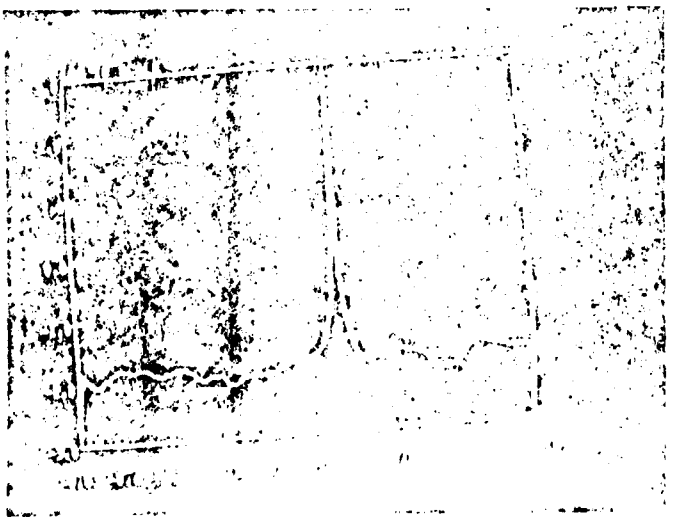


Fig. 3f Maximum eigenvalues of spectrum for complex signal.

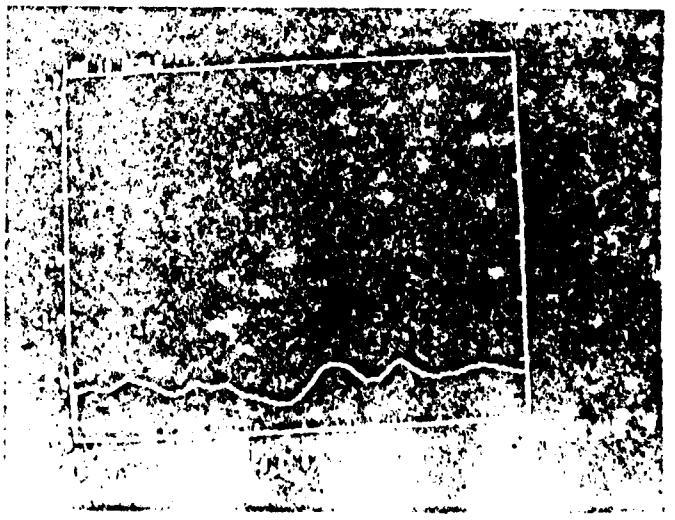


Fig. 3g Minimum eigenvalues of spectrum for complex signal.

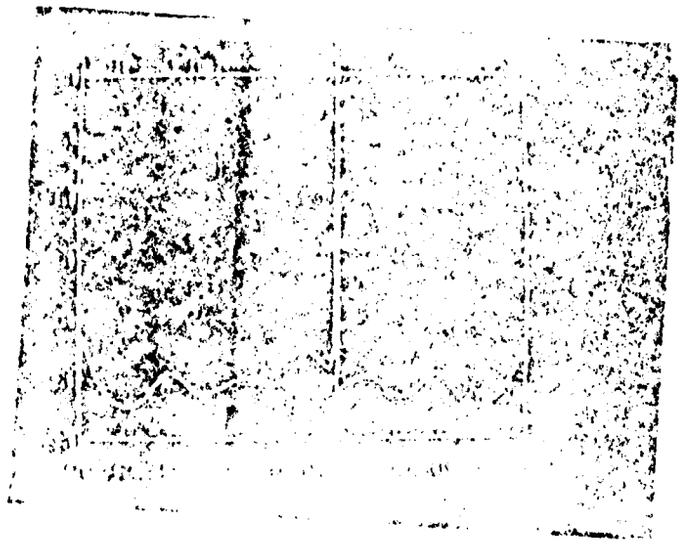


Fig. 4a 1-1 spectrum, $\gamma_{11}(f)$ for complex signal.

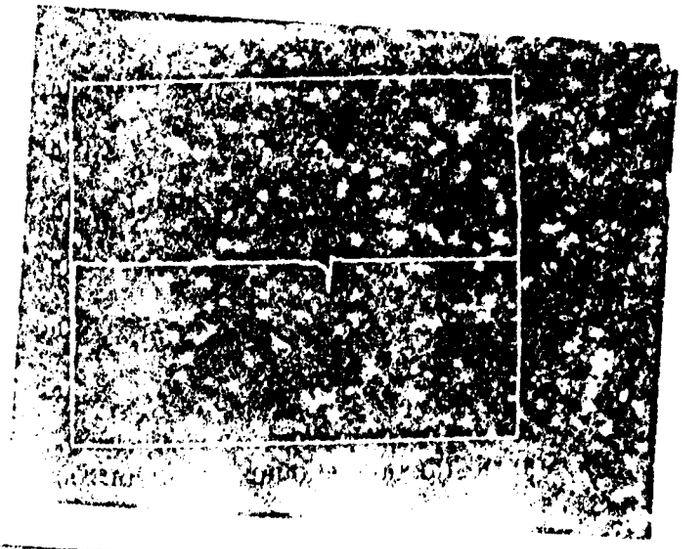


Fig. 4b Phase of 1-2 spectrum, are $\gamma_{12}(f)$ for complex signal. Given freq.=0.0[illegible] Hz and phase diff.=0.[illegible] rad.

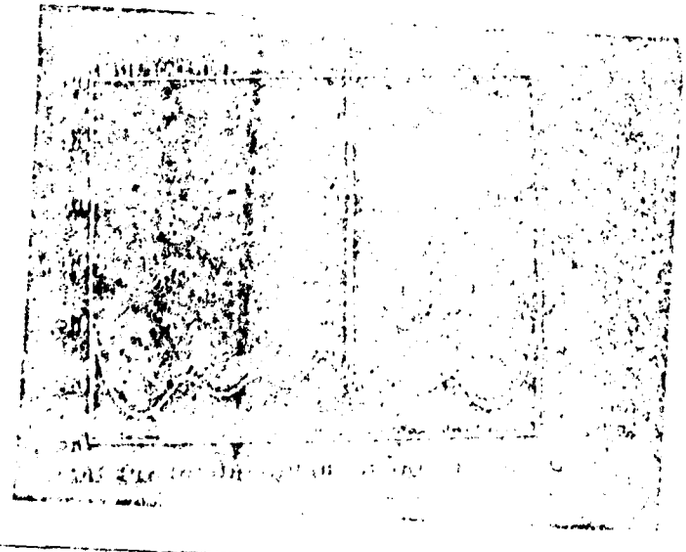


Fig. 5a 1-1 spectrum, $\gamma_{11}(f)$ for complex signal.

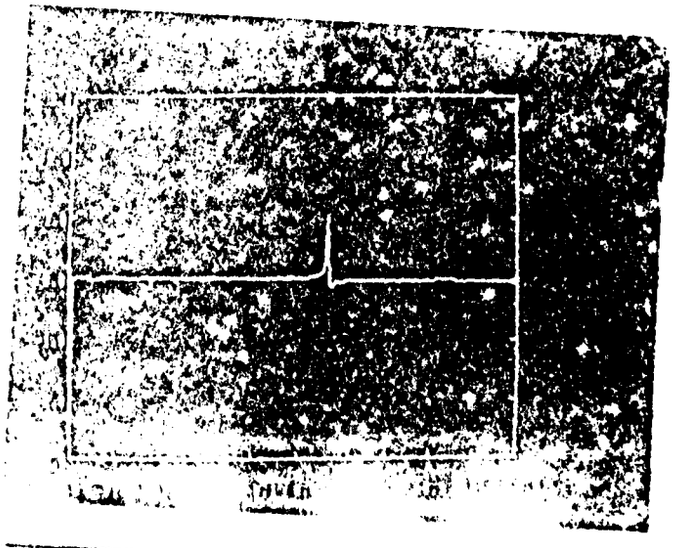


Fig. 5b Phase of 1-2 spectrum, are $\gamma_{12}(f)$ for complex signal. Given freq.=0.0625 Hz and phase diff. = -1 rad.

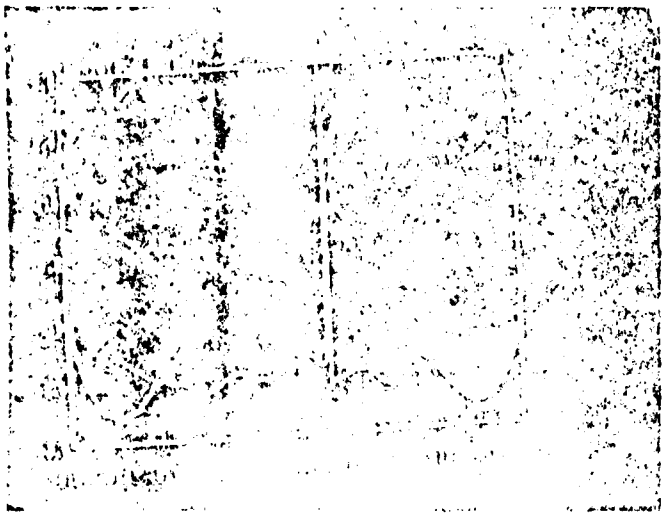


Fig. 6a 1-1 spectrum, $\Gamma_{11}(f)$ for complex signal.

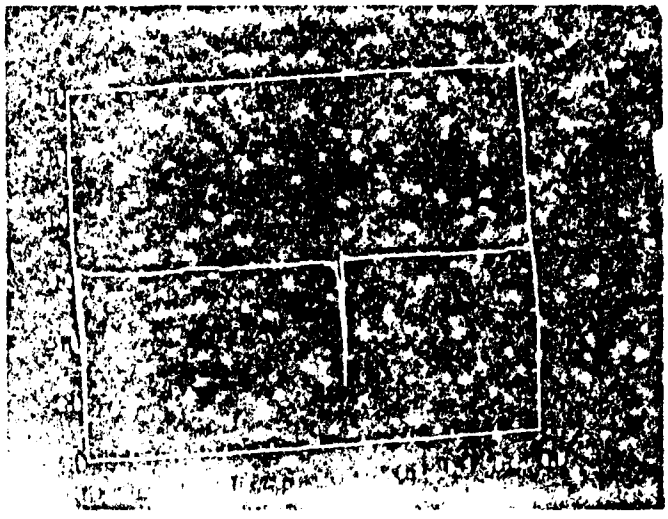


Fig. 6b Phase of 1-2 spectrum, arg $\Gamma_{12}(f)$ for complex signal. Given freq.=0.1.5, and phase diff.=2 rad.

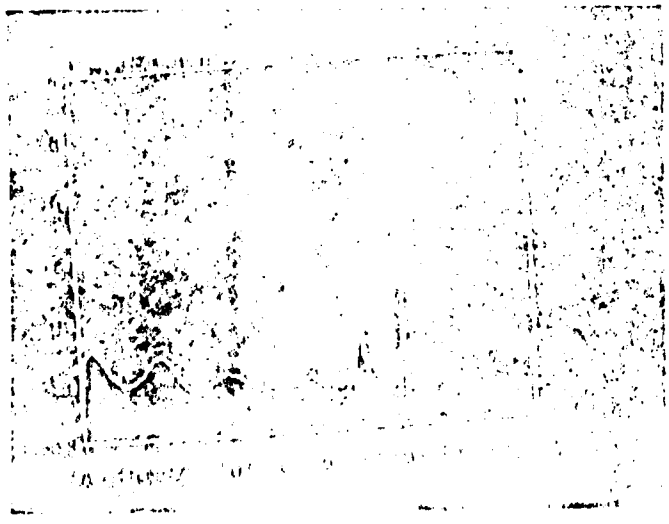


Fig. 7a 1-1 spectrum $\Gamma_{11}(f)$ for complex signal.

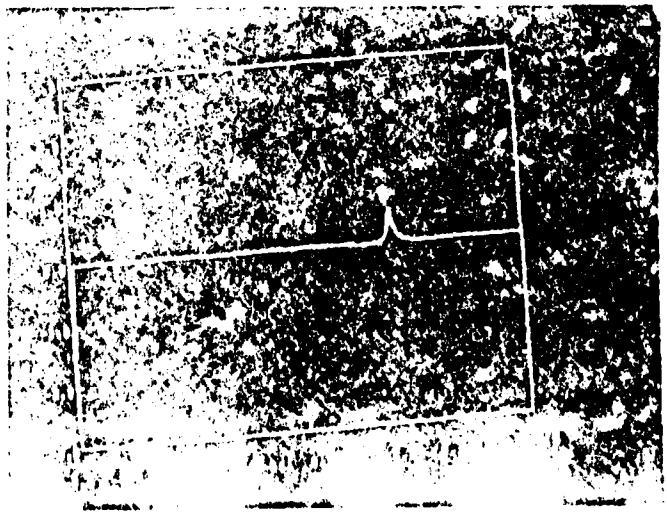


Fig. 7b Phase of 1-2 spectrum, arg $\Gamma_{12}(f)$ for complex signal. Given freq.=0.1 Hz. and phase diff.=-0.5 rad.

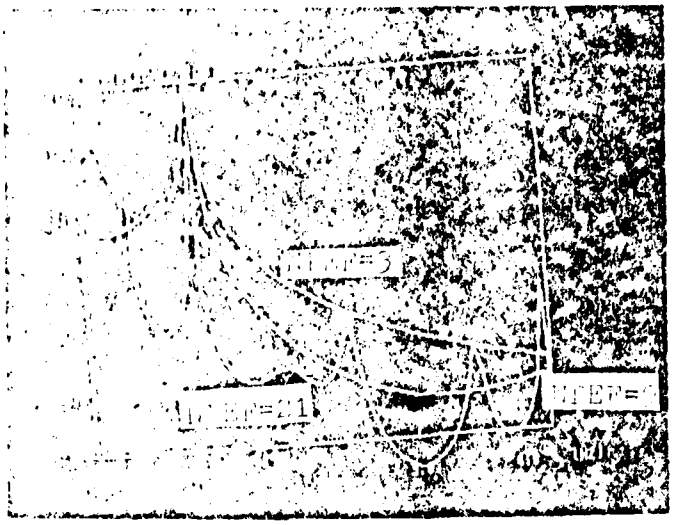


Fig. 5a Maximum entropy power spectra of single channel real data. (ϕ_1=0, λ_1=0.95, Nd=24, NPEF=3, 9, 21).

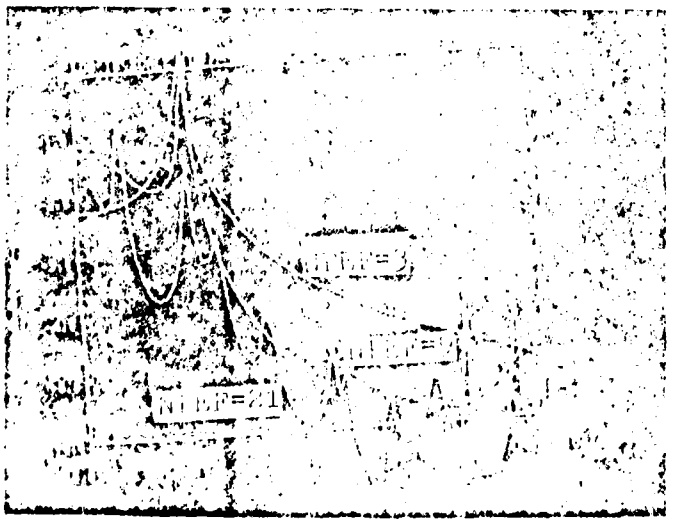


Fig. 5b Maximum entropy power spectra of 2-channel real data. (ϕ_1=0, λ_1=1.0, λ=0.05, N_d=24, NPEF=3, 9, 21)

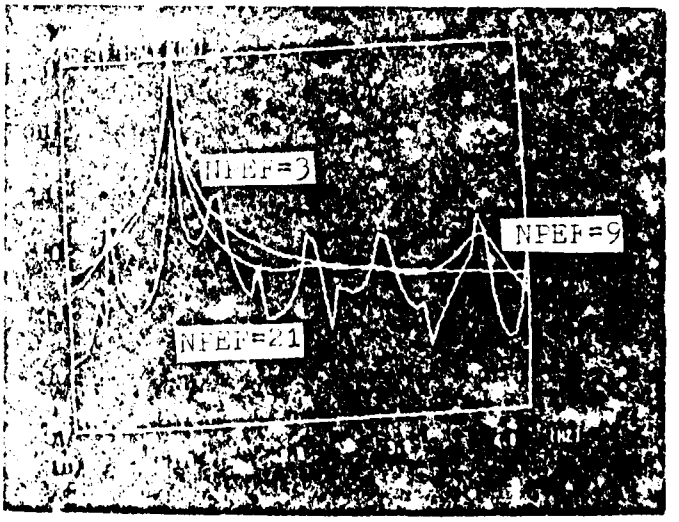
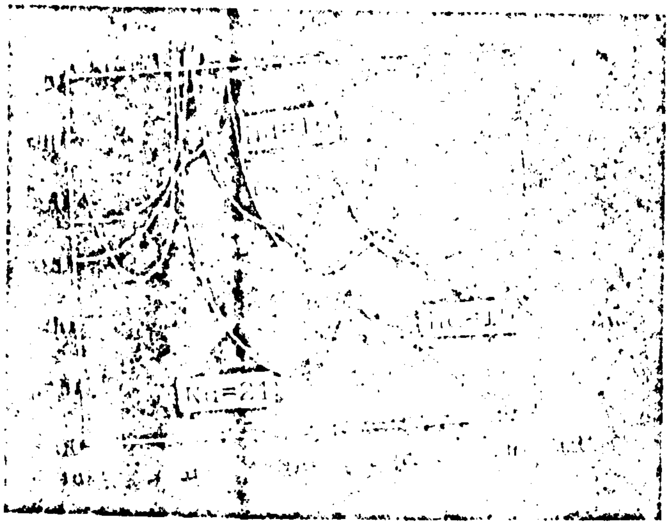


Fig. 5c Same as Fig. 5b but with complex signals.



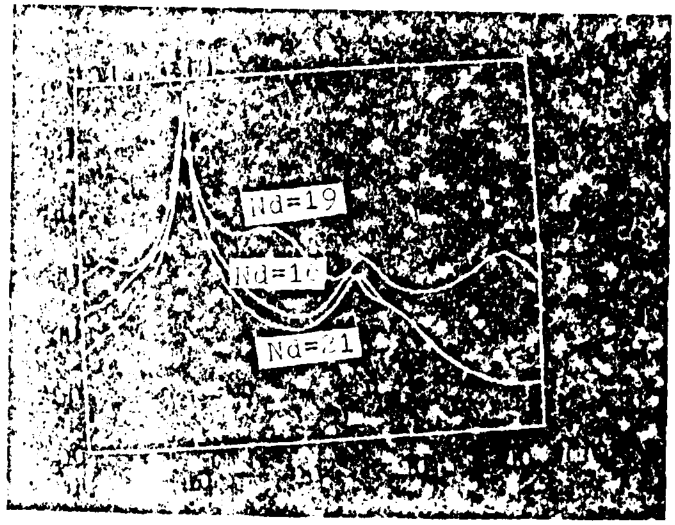
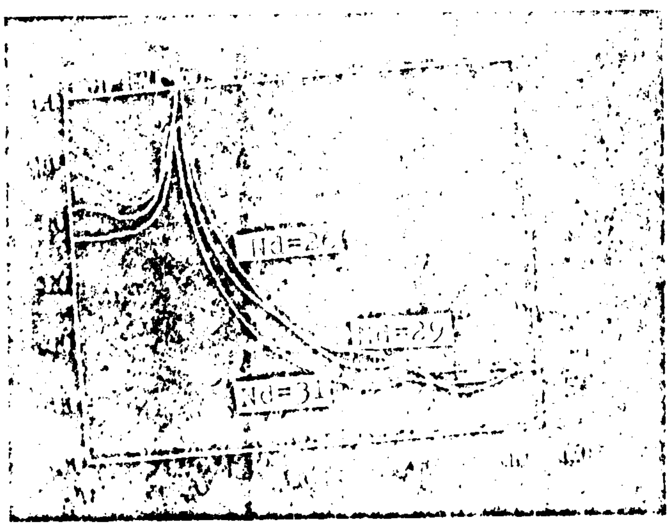


Fig. 9a Maximum entropy power spectra for [illegible]-channel real data.
(δ_1=0, [illegible],
NREF=9, Nd=[illegible])

Fig. 9b Same as Fig. 9a but with complex signals.



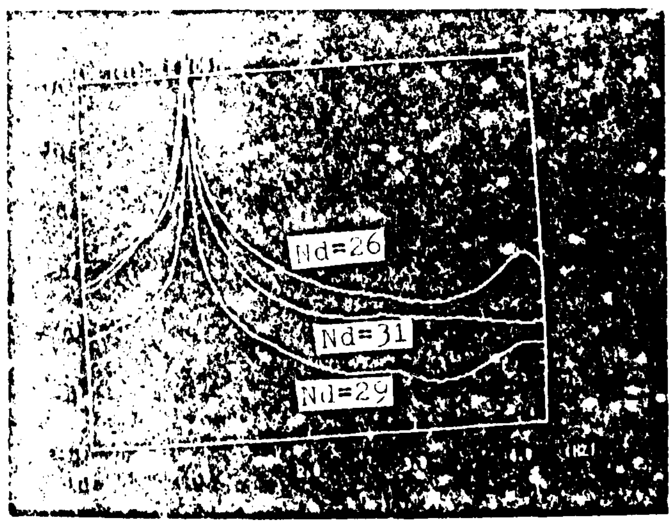


Fig. 10a Maximum entropy power spectra for [illegible]-channel real data.
(δ_1=0, [illegible], [illegible]0.05,
NREF=9, Nd=[illegible])

Fig. 10b Same as Fig. 10a, but with complex signals.

Fig. 11a Maximum entropy power spectrum of 2-channel real data. (θ_1=0, θ_2=0, A=0.95, M=2, Nd=30, 35, 41)

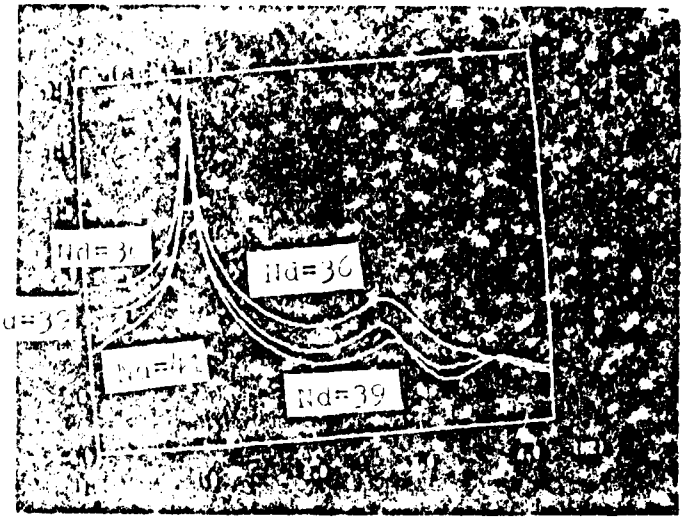


Fig. 11b Same as Fig. 11a but with complex signals.

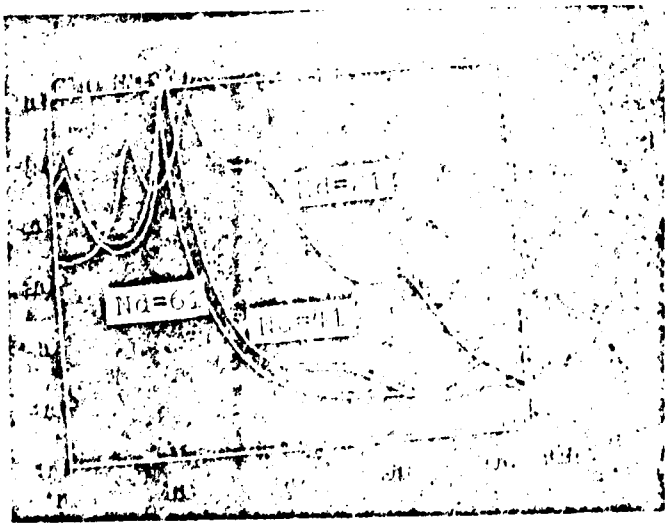


Fig. 12a Maximum entropy power spectra of 2-channel real data. (θ_1=45, θ_2=135, A=1.05, M=2, Nd=21, 41, 61)

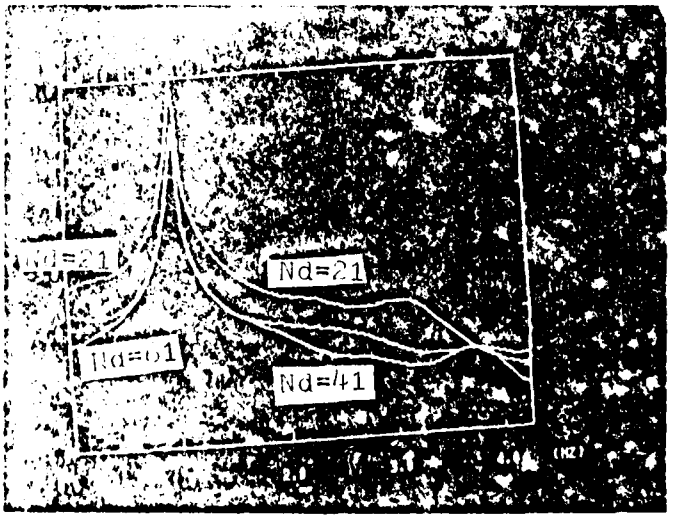


Fig. 12b Same as Fig. 12a but with complex signals.

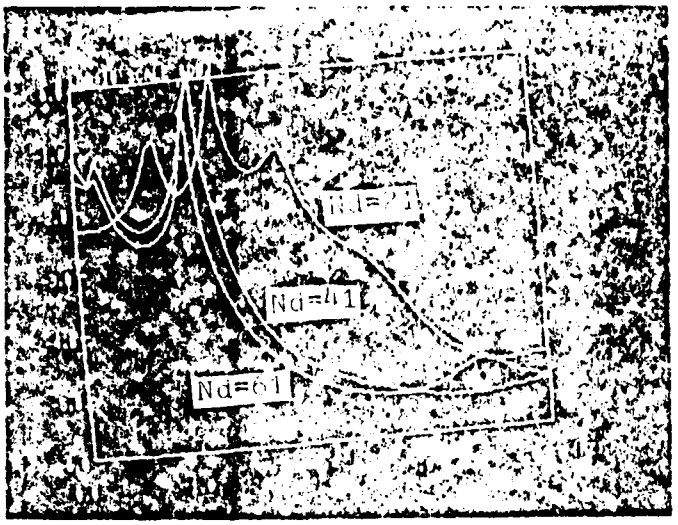


Fig. 13a Maximum entropy power spectra of 2-channel real data. (ϕ_1=-45, ϕ_2=-135, A=0.05, NPEF=9, Nd=21,41,61)

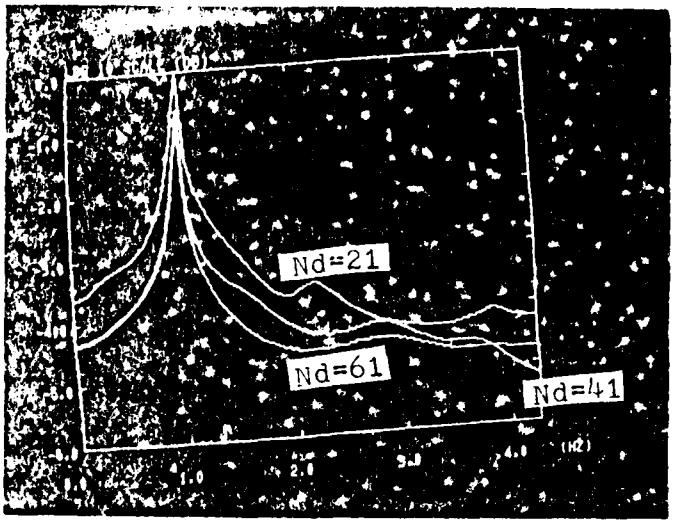


Fig. 13b Same as Fig. 13a but with complex signals.

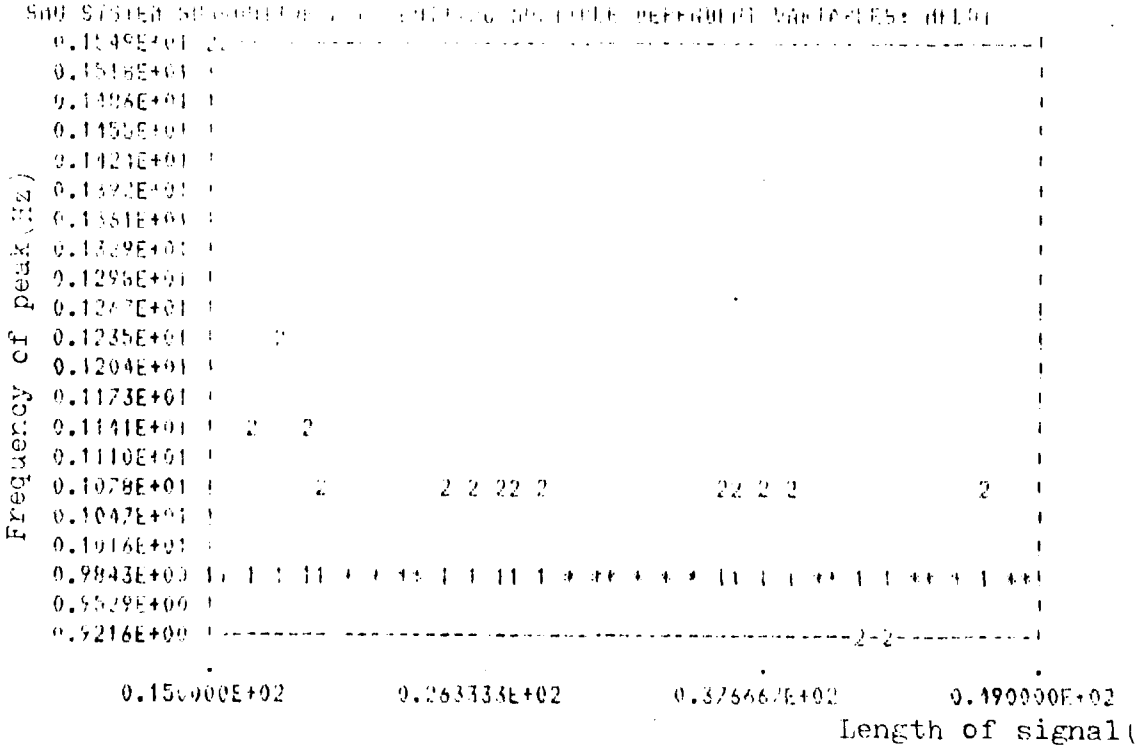


Fig. 14 The location of spectral peak for various lengths of 1Hz sinusoids. "1" denotes complex signals, "2" denotes real signals and "*" denotes both "1" and "2". (ϕ_1=0, ϕ_2=130, A=0.05, NPEF=9)

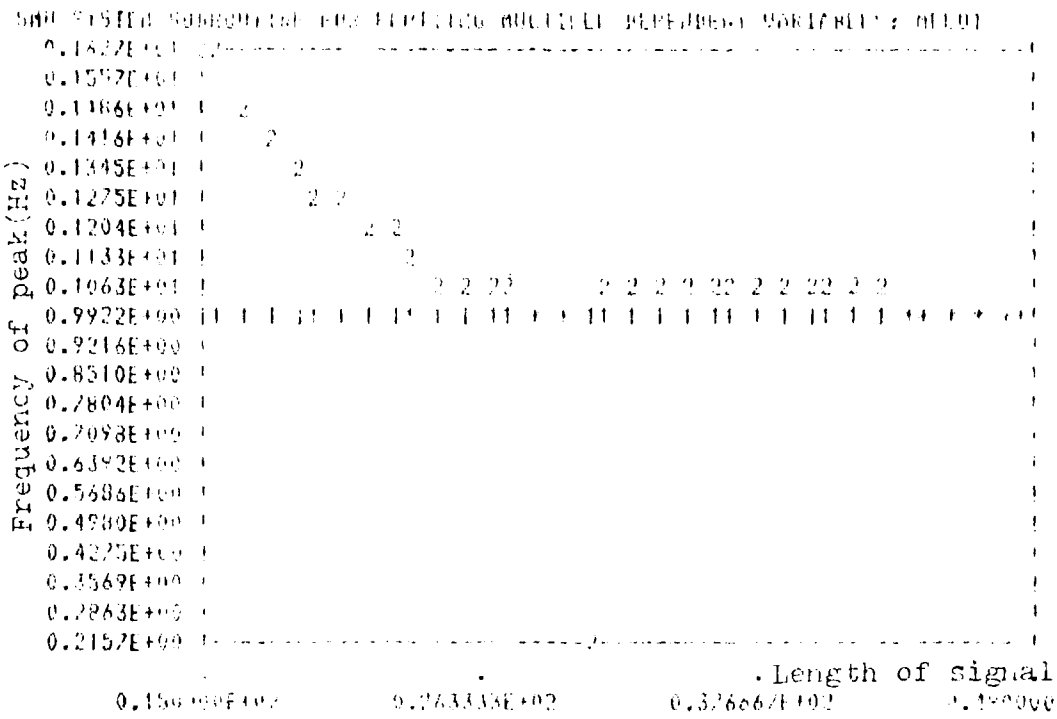


Fig. 15 The location of spectral peak for various lengths of 1Hz sinusoids. "1" denotes complex signals, "2" denotes real signals, and "*" denotes both "1" and "2" (Φ_1=45, Φ_2=135, A=0.05, NPEF=9)

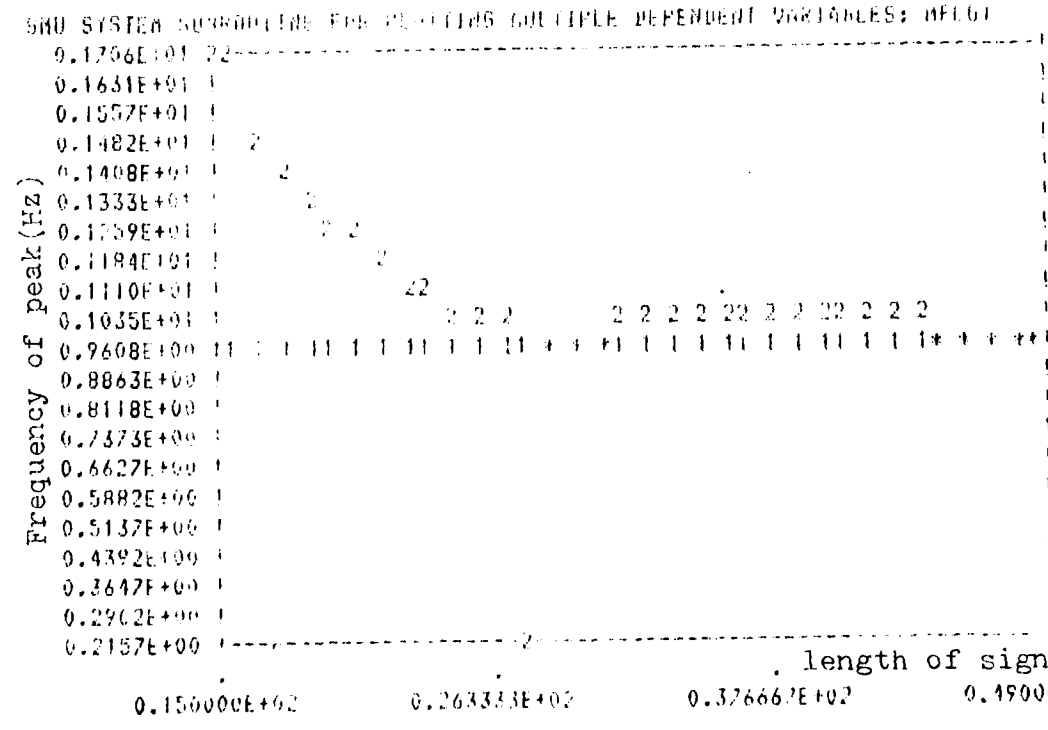


Fig. 16 Same as Fig. 14 but with Φ_1=-45, Φ_2=-135, A=0.05, NPEF=9.

<u>Appendix</u> Computer Program for the Multichannel Channel Maximum Entropy Spectral Analysis

```
 TYPE [illegible]
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C       MULTICHANNEL MAXIMUM ENTROPY ROUTINE
C       NORG (GENERAL) P-CHANNEL, COMPLEX.
C       ARGUMENTS:
C          NC    MAX. FILTER LENGTH.
C          NP    # OF CHANNELS.
C          NSPE MIN. FILTER LENGTH(DESIRED SPECTRUM).
C          NDAT # OF DATA POINTS.
C          NPRS MIN. FILTER LENGTH(DESIRED PRINTOUT).
C
C       INPUT DATA   T STORED IN FILE  SIN.DAT'.
C
C       THIS PROGRAM MUST LINK WITH THE FOLLOWING SUBROUTINES:
C          DRAW         PROGRM TO DISPLAY THE DESIRED DATA
C                       ON THE TERMINAL.
C          [illegible]
C          EIGCH        SUBROUTINES OF IMSL SOFTWARE PACKAGE.
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      COMMON/[illegible]/YT,DT,NDAT,NP
      COMMON/[illegible]/NC
      COMMON/[illegible]
      COMMON/[illegible]/NPRS
      DIMENSION YT(4,300)
      DIMENSION UU(4),VV(4)
      COMPLEX YT
        TYPE 1110
 1110   FORMAT(2X,'I/P NC,NP,NSPE,NDAT,NPRS,DT')
        ACCEPT *,NC,NP,NSPE,NDAT,NPRS,DT
      TYPE 4
    4 FORMAT(/)
      TYPE 15,NC,NP
   15 FORMAT(1X,27HMAX. NO. OF FILTER COEFF. =,I15,2X,41HNO. OF COMPONE
     1NTS IN VECTOR TIME SERIES =,I15,//)
      TYPE 7,NDAT,DT
    7 FORMAT(1X,24HORIGINAL TIME SERIES HAS,I17,2X,28HDATA POINTS AT TI
     1ME INTERVAL,1F10.5,/)
        OPEN(UNIT=30,FILE='SIN.DAT')
      DO 2 K=1,NDAT
      READ (30,20)(UU(I),VV(I),I=1,NP)
   20 FORMAT(4F10.3)
      DO 3 I=1,NP
    3 YT(I,K)=CMPLX(UU(I),VV(I))
    2 CONTINUE
      CALL DEIGEN
      CALL CALFIL
        CALL EXIT
      END
```

```
C*************************************************************
      SUBROUTINE DETREN
      COMMON/ETW/ I,91,NDAT,IT
      DIMENSION (ETA,9..) X(300),Y(300)
      COMPLEX Z1,EMAX,V10,V11,V12,V13,EAT,A
      ENDAT=FLOAT(NDAT)
      TYPE 1
    1 FORMAT(/)
      TYPE 2,NDAT,DT
    2 FORMAT(1X,33HNO. OF DATA POINTS ACTUALLY USED =,I10,3X,33HWITH R
     1ESULTING TIME INTERVAL DELTA T =,1E12.10,/)
      TYPE 3,N
    3 FORMAT(1X,30HTHIS COMPLEX VECTOR TIME SERIES HAS ,I15,2X,10HCOMPO
     1NENTS,/)
        V1=(0.,1.0)*12.
        V2=ENDAT+1.
        V2=V2*(ENDAT-1.)
        V3=V1*ENDAT
      V12=V1/V2
      DO 4 K=1,N
      TYPE 1
      TYPE 6,K
    6 FORMAT(1X,33HTHE DATA BELOW PERTAINS TO COMPONENT NO. ,I15,3X,14H
     1OF TIME SERIES,//)
      EAT=CMPLX(0.,0.)
      DO 5 I=1,NDAT
        X(I)=REAL(Z(K,I))
        Y(I)=AIMAG(Z(K,I))
    5 EAT=EAT+Z(K,I)
      EAT=EAT/ENDAT
      TYPE 10
   10 FORMAT(1X,50HREAL PARTS OF ORIGINAL TIME SERIES COMPONENT BELOW)
      TYPE 1
      TYPE 11,(X(I),I=1,NDAT)
   11 FORMAT(10F12.2)
      TYPE 1
      TYPE 12
   12 FORMAT(1X,55HIMAGINARY PARTS OF ORIGINAL TIME SERIES COMPONENT BE
     1LOW,/)
      TYPE 11,(Y(I),I=1,NDAT)
      TYPE 1
        EAYR=REAL(EAT)
        EAYI=AIMAG(EAT)
      TYPE 7,EAYR,EAYI
    7 FORMAT(1X,52HMEAN OF THIS COMPONENT OF RAW TIME SERIES HAS REAL P
     1ART =,1E15.7,2X,20HAND IMAGINARY PART =,1E15.7,/)
C  IF THE TWO CARDS ISKIP=1 AND IF(ISKIP.EQ.1)GO TO 30 ARE PLACED
C  BETWEEN THE 1,5 BELOW,THE DATA WILL NOT BE DETRENDED. IF THEY ARE
C  PLACED BETWEEN THE 2,5,THE MEAN WILL BE SUBTRACTED.  IF BETWEEN
C  THE 3,5,THE DATA WILL BE GIVEN A FULL LINEAR DETREND.
C 1 1 1 1 1 1 1 1 1 1 1 1 1 NO DETREND 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
      ISKIP=1
      IF(ISKIP.EQ.1)GO TO 30
C 1 1 1 1 1 1 1 1 1 1 1 1 1 NO DETREND 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
C         [illegible]
  [illegible] FORMAT(5X,[illegible] HAS BEEN SUBTRACTED FROM EACH VALUE [illegible] THIS COM
   1PONENT,/)
     [illegible] 1,[illegible]
     [illegible]
     [illegible]
   9 [illegible]
     TYPE [illegible]
  13 FORMAT(1X,5[illegible]REAL PARTS OF MEAN-REDUCED TIME-SERIES COMPONENT BE
   1LOW",/)
     TYPE 11,(X(I),I=1,NDAT)
     TYPE 1
     TYPE 14
  14 FORMAT(1X,5[illegible]IMAGINARY PARTS OF MEAN-REDUCED TIME-SERIES COMPONEN
   1T BELOW,/)
     [illegible] 11,([illegible](I),I=1,NDAT)
     TYPE 1
C [illegible] 2 2 2 2 [illegible] 2 SUBTRACT MEAN 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
C [illegible] 2 2 [illegible] 2 SUBTRACT SE[illegible] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
         [illegible]
      [illegible]
     DO 15 [illegible]
     [illegible]
  15 [illegible]
       [illegible]
       [illegible]
       [illegible]
     TYPE 16,[illegible]
  16 FORMAT(1X,[illegible]REGRESSION COEFFICIENT FOR DETRENDING, REAL PART =,1
   1E12.6,[illegible]IMAGINARY PART =,1E12.6,/)
     TYPE [illegible]
     [illegible]
     DO 17 [illegible]
     [illegible]
       [illegible]
       [illegible]
  17 [illegible]
     TYPE 1
     TYPE 18
  18 FORMAT(1X,[illegible]DETRENDED TIME SERIES COMPONENT BELOW,/)
     TYPE 19
  19 FORMAT(1X,5[illegible]REAL PARTS OF DETRENDED TIME SERIES COMPONENT BELOW,
   1/)
     TYPE 11,(X(I),I=1,NDAT)
     TYPE 20
  20 FORMAT(1X,5[illegible]IMAGINARY PARTS OF DETRENDED TIME-SERIES COMPONENT B
   1ELOW,/)
     TYPE 11,(Y(I),I=1,NDAT)
     TYPE 21
  21 FORMAT([illegible]DETRENDED TIME-SERIES COMPONENT USED IN SUBSEQUENT C
   1ALCULATIONS,/)
C [illegible] 3 3 3 3 3 3 3 3 3 3 3 3 3FULL DETREND 3 3 3 3 3 3 3 3 3 3 3 3
C 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3FULL DETREND 3 3 3 3 3 3 3 3 3 3 3 3 3
  50 CONTINUE
     TYPE 1
   4 CONTINUE
     TYPE 1
     RETURN
     END
```

```
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
      SUBROUTINE CALCUL
      COMMON/CT/YI,ET,NDAT,NP
      COMMON/DAT/GI
      COMMON/SPECTC/NSEC
      COMMON/UF1/BI,YI,CR,CF,C,CP,E,R,G,SS,CN,CNP,PI,EI,RRB,IS,CNDAT,WA
     1,NN,C,NI,NPP,F,PP,NAI
      COMMON/PROB/FLL,ELPP,FPE,DETP,DETPP
      COMMON/EIGEN/AEIG,WK,EINV
      COMPLEX YI,BI,EI,CR,CF,C,CP,E,B,G,SS,CN,CNP,PI,EI,RRB,V1,V2,V3,CN
     1DAT,CN,F,PP,AEIG,EINV
      DIMENSION YI(4, 300),BI(4, 300),EI(4, 300),CR(4,4,40),CF(4,4,40),
     1C(4,4),CP(4,4),E(4,4),B(4,4),G(4,4),SS(4,4),CN(4,4),CNP(4,4),BI(4,
     2 4),EI(4,4),RRB(16),V1(4,4),V2(4,4),V3(4,4),FR1(4,4),FR2(4,4),F(4,4
     3),PP(4,4),AEIG(4,4),WK(12),EINV(16,16)
      DIMENSION EL(4),ELP(40,2),ELPP(40,2),FPE(40),DETP(40),DETPP(40)
        NG=NDAT
        LNDAT=IFIX(NDAT)
        CNDAT=CMPLX(CNDAT,0.)
        IS=1
        NA1=NDAT-1
        NPP=NP
      DO 1 I=1,NP
      DO 2 J=1,NP
      DO 3 K=1,40
      CR(I,J,K)=CMPLX(0.,0.)
    3 CF(I,J,K)=CMPLX(0.,0.)
      C(I,J)=CMPLX(0.,0.)
    2 CP(I,J,1)=CMPLX(0.,0.)
      C(I,I)=CMPLX(1.,0.)
      CP(I,I)=CMPLX(1.,0.)
    1 RRB(I)=CMPLX(0.,0.)
      DO 23 I=1,NP
      DO 23 J=1,NP
      SS(I,J)=CMPLX(0.,0.)
      DO 4 K=1,NDAT
    4 SS(I,J)=SS(I,J)+YI(I,K)*CONJG(YI(J,K))
   23 SS(I,J)=SS(I,J)/CNDAT
      DO 22 K=1,NDAT
      EI(I,K)=YI(I,K)
   22 BI(I,K)=YI(I,K)
      DO 24 I=1,NP
      DO 24 J=1,NP
      F(I,J)=SS(I,J)
   24 PP(I,J)=SS(I,J)
      DO 34 I=1,NP
      DO 34 J=1,NP
   34 AEIG(I,J)=SS(I,J)
      CALL VCVTCS(AEIG,NP,4,AEIG)
      CALL EIGCH(AEIG,NP,0,EL,V1,4,WK,IER)
        ELP(1,1)=EL(NP)
        ELPP(1,1)=EL(NP)
        ELP(1,2)=EL(1)
      ELPP(1,2)=EL(1)
      DETP(1)=1.
      DO 35 I=1,NP
   35 DETP(1)=DETP(1)*EL(I)
      DO 5 I=1,NP
      DO 5 J=1,NP
      FR1(I,J)=REAL(SS(I,J))
    5 FR2(I,J)=AIMAG(SS(I,J))
      IN=0.
```

```
      [illegible] 1,3
    6 [illegible]
      CALL [illegible]
      [illegible]
      TYPE 8
    8 FORMAT(/)
      TYPE 7
    7 FORMAT(10X,29HOVERALL VARIANCE MATRIX OF [illegible]/)
      TYPE 9
    9 FORMAT(4(6X,4HREAL,10X,5HIMAG.,3X))
      TYPE 10,((PF1(I,J),PF2(I,J),J=1,NF),I=1,NF)
   10 FORMAT(8E15.6)
      TYPE 11,TR
   11 FORMAT(10X,7HTRACE =,1E20.10,//)
      DO 12 L=1,NFF
      IP=L
      CALL [illegible]
      CALL [illegible]
      IF(L.LT.NSPE)GO TO 33
      CALL SPECAL
   33 CONTINUE
   12 CONTINUE
      RETURN
      END
C*************************************************************************
      SUBROUTINE SPECAL
      COMMON/[illegible]/IT,DT,NDAT,NP
      COMMON/[illegible]/NFF
      COMMON/EIGIN/NEIG,UK,EINV
      COMMON/[illegible](256),N2,N3,W(19)
      COMMON/[illegible]/F,FI,CB,CF,C,CP,Z,B,G,SS,[illegible],BI,EI,BBB,IP,CMDAT,WM
     1,[illegible],FI,[illegible],F,FF,NNI
      COMPLEX [illegible],B,CF,C,CP,E,B,G,SS,[illegible],BI,EI,BBB,V11,V21,V33
     1,V44,V55,V66,CMDAT,WM,P,FP,FFS,V1,EINV,NEIG
      COMPLEX [illegible]
      DIMENSION Y([illegible], 300),[illegible](4, 300),[illegible](4, 300),[illegible](4,4,40),[illegible](4,4,40),
     1[illegible],E(4,4),B(4,4),G(4,4),SS(4,4),[illegible](4,4),CMP(4,4),BI(4,
     2[illegible]),EI(4,4),BBB(16),V11(4,4),V21(4,4),V33(4,4),V44(4,4),V55(4,4),V6
     36(4,4),P(4,4),FP(4,4),S(256,4),FREQ(256),FFS(4,4,256),EL(4),V1(4,4
     4),[illegible],NEIG(4,4),[illegible]
      DIMENSION [illegible](40,2),ELPP(40,2),FFE(40),DEIF(40),DEIFF(40)
      COMMON/[illegible]/[illegible],ELPP,FFE,DEIF,DEIFF
      COMMON/[illegible]/XLOWER,XUPPER,YLOWER,YUPPER,KDIVX,KDIVY,KSCLX,KSCLY
      COMMON/[illegible]/[illegible](8),LAST(8),ID(8)
      COMMON/[illegible](2),YSUB(2)
      COMMON/[illegible],NFR
        [illegible]
        [illegible]
        [illegible]
        [illegible]
        [illegible]
        [illegible]
        EN2=LOG([illegible])
        [illegible]=[illegible]/(EN2*DT)
      DO 21 I=1,NF
      DO 21 [illegible]
      DO 25 K=1,[illegible]
   25 [illegible](K)=CMPLX(0.,0.)
      DO 26 K=1,[illegible]
   26 [illegible](K)=CMPLX([illegible])
      CALL [illegible]
```

```
      DO 2? K=1,N?
   2? FFS(I,J,K)=A(K)
      DO ?? K=1,N?
      DO ?? I=1,NP
      DO 28 J=1,NP
   28 EINV(I,J)=FFS(I,J,K)
      CALL CAIGV(EINV,NP,BFB)
      DO 29 I=1,NP
      DO 29 J=1,NP
      V33(I,J)=CMPLX(0.,0.)
      DO 29 KK=1,NP
   29 V33(I,J)=V33(I,J)+EINV(I,KK)*P(KK,J)
      DO 30 I=1,NP
      DO 30 J=1,NP
      V44(I,J)=CMPLX(0.,0.)
      DO 31 KK=1,NP
   31 V44(I,J)=V44(I,J)+V33(I,KK)*CONJG(EINV(J,KK))
   30 FFS(I,J,K)=V44(I,J)*DT
   27 CONTINUE
      DO 36 K=1,N3P1
      AKM1=FLOAT(K-1)
   36 FREQ(K)=[illegible]*AKM1
      DO 37 K=1,N3
        [illegible]=FLOAT(K-1-N3)
        N3PK=N3+K
   37 FREQ(N3PK)=DF*[illegible]
      DO 60 K=1,N3
        N3PK=N3+K
        XX1=FREQ(K)
        XX2=FREQ(N3PK)
      DO 61 I=1,NP
      DO 61 J=1,NP
        V55(I,J)=FFS(I,J,K)
        V66(I,J)=FFS(I,J,N3PK)
        FFS(I,J,K)=V66(I,J)
   61 FFS(I,J,N3PK)=V55(I,J)
      FREQ(N3PK)=XX1
   60 FREQ(K)=XX2
      TYPE 33
   33 FORMAT(/)
      DO 40 K=1,N2
      DO 40 I=1,NP
   40 S(K,I)=REAL(FFS(I,I,K))
      DO 50 I=1,NP
      TYPE 39,I,I
   39 FORMAT(1X,I1,1H.,I1,21H-POWER SPECTRUM BELOW,/)
        III=1
        CALL DRAW(S,III,I)
      TYPE 41,(S(K,I),K=1,N2)
   41 FORMAT(10E13.5)
      TYPE 42
   50 CONTINUE
      TYPE 42
   42 FORMAT(/)
      IF(NP.EQ.1)GO TO 51
      NPM1=NP-1
      DO 82 I=1,NPM1
      IP1=I+1
```

```
      DO 32 J=I+1,NP
      DO 33 K=1,N2
        XX1=REAL(FFS(I,J,K))
        XX2=AIMAG(FFS(I,J,K))
      S(K,I)=CABS(FFS(I,J,K))
33    S(K,J)=ATAN2(XX2,XX1)
      TYPE 42
      TYPE 44,I,J
44    FORMAT(1X,14HABS. VALUE OF ,I2,1H,,I2,21H-POWER SPECTRUM BELOW,/)
        III=I
        CALL DRAW(S,III,1)
      TYPE 41,(S(K,I),K=1,N2)
      TYPE 42
      TYPE 45,I,J
45    FORMAT(1X,18HPHASE ANG. FOR,I2,1H,,I2,21H-POWER SPECTRUM BELOW,/)
        III=J
        CALL DRAW(S,III,2)
      TYPE 41,(S(K,J),K=1,N2)
      TYPE 42
      DO 55 K=1,N2
        XX1=REAL(FFS(I,I,K))
        XX2=REAL(FFS(J,J,K))
55    S(K,I)=(CABS(FFS(I,J,K)))**2)/(XX1*XX2)
      TYPE 56,I,J,I,I,J,J
56    FORMAT(1X,25HSQUARED COHERENCE =(ABS(S,2I1,2H)**2)/S,2I1,2H*S,2I1
     1,6H BELOW,/)
        CALL DRAW(S,I,2)
      TYPE 41,(S(K,I),K=1,N2)
32    CONTINUE
31    CONTINUE
      TYPE 46
46    FORMAT(5X,20HFREQUENCIES BELOW,/)
      TYPE 41,(FREQ(K),K=1,N2)
      TYPE 42
      DO 47 K=1,N2
      DO 48 I=1,NP
      DO 48 J=1,NP
48    AEIG(I,J)=FFS(I,J,K)
      CALL VCVTCH(AEIG,NP,4,AEIG)
      CALL EIGCH(AEIG,NP,0,EL,Z1,4,WK,IER)
      S(K,1)=EL(NP)
      S(K,2)=EL(1)
47    CONTINUE
      TYPE 49
49    FORMAT(1X,32HMAXIMAL EIGENVALUES OF SPECTRUM BELOW,/)
        CALL DRAW(S,1,1)
        TYPE 41,(S(K,1),K=1,N2)
      TYPE 42
      TYPE 50
50    FORMAT(1X,32HMINIMAL EIGENVALUES OF SPECTRUM BELOW,/)
        CALL DRAW(S,2,1)
      TYPE 41,(S(K,2),K=1,N2)
      TYPE 38
      TYPE 70
70    FORMAT(5X,35HDETERMINANTS OF FORWARD POWER BELOW,/)
      TYPE 41,(DETP(K),K=1,IFP1)
      TYPE 42
```

```
      DO 22 K=1,NFPE
   22 FLLD(K)=FLOAT(K-1)
      TYPE 41
      TYPE 172
  172 FORMAT(1X,[illegible]HAKAIKE FPE CRITERIA BELOW,/)
      TYPE 41,(FPE(I),I=1,NPI)
      TYPE 42
      RETURN
      END
C*****************************************************************
      SUBROUTINE UPRINT
      COMMON/BPB/BT,ET,CP,CF,C,CP,E,B,G,SS,CN,CNP,BI,EI,PPD,I9,CNDAT,WN
     1,NN,M,N1,NPP,P,PP,NMI
      COMMON/POWEL/ELP,ELPP,FPE,DETP,DETPP
      COMMON/NPSERI/NPRS
      COMMON/EIGEN/AEIG,WK,EINV
      COMPLEX BT,ET,CF,CP,C,CP,E,B,G,SS,CN,CNP,BI,EI,RRB,V1,V2,V3,CNDAT
     1,WN,CNNI,CI,P,PP,AEIG,EINV
      DIMENSION BT(4,300),ET(4,300),CB(4,4,40),CF(4,4,40),C(4,4),CP(4
     1,4),E(4,4),B(4,4),G(4,1),SS(4,4),CN(4,4),CNP(4,4),BI(4,4),EI(4,4),
     2 RRB(16),V1(4,4),V2(4,4),V3(4,4),P(4,4),PP(4,4),AEIG(4,4),EINV(16,1
     36)
      DIMENSION VRR(4,300),VRI(4,300),VAR(4,1),VAI(4,4),EL(4),WK(12)
      DIMENSION ELP(40,2),ELPP(40,2),FPE(40),DETP(40),DETPP(40)
        [illegible]
        [illegible]
        [illegible]
        [illegible]
        [illegible]
      IF(I9.LT.NPRS)GO TO 50
      TYPE 1
    1 FORMAT(/)
      TYPE 2,I9,I9+1
    2 FORMAT(1X,15HRESULTS BELOW FOR ,I15,3X,42HLAG(S),I.E. MATRIX OF P
     1IECEWISE DIMENSION ,I15,1X,I5,/)
      TYPE 3
    3 FORMAT(10X,23HFORWARD RESIDUALS BELOW,/)
      DO 4 I=1,N
      DO 4 J=1,NNI
      VRR(I,J)=REAL(ET(I,J))
    4 VRI(I,J)=AIMAG(ET(I,J))
      DO 7 I=1,N
      TYPE 5,I
    5 FORMAT(5X,16HCOMPONENT NUMBER ,I15,2X,5HBELOW,/)
      TYPE 6
    6 FORMAT(20X,16HREAL PARTS BELOW,/)
      TYPE 8,(VRR(I,J),J=1,NNI)
    8 FORMAT(10F13.8)
      TYPE 9
    9 FORMAT(/)
      TYPE 10
   10 FORMAT(1X,21HIMAGINARY PARTS BELOW,/)
      TYPE 8,(VRI(I,J),J=1,NNI)
      TYPE 1
    7 CONTINUE
      TYPE 50
   50 FORMAT(10X,24HBACKWARD RESIDUALS BELOW,/)
```

```
      DO 43 I=1,NP
      DO 43 J=1,NM1
      V4R(I,J)=REAL(B(I,J))
   43 V4I(I,J)=AIMAG(B(I,J))
      DO 49 I=1,NP
      WRITE (7,1
      FILE 7
      WRITE (7,(V4R(I,J),J=1,NM1)
      TYPE 9
      TYPE 10
      TYPE 8,(V4I(I,J),J=1,NM1)
      TYPE 1
   49 CONTINUE
      TYPE 11
   11 FORMAT(10X,14HE-MATRIX BELOW,/)
      DO 12 I=1,NP
      DO 12 J=1,NP
      V4R(I,J)=REAL(E(I,J))
   12 V4I(I,J)=AIMAG(E(I,J))
      TYPE 3
      TYPE 13,((V4R(I,J),J=1,NP),I=1,NP)
   13 FORMAT(4E20.10)
      TYPE 10
      TYPE 13,((V4I(I,J),J=1,NP),I=1,NP)
      TYPE 9
      IR=0.
      DO 46 I=1,NP
   46 IR=IR+V4R(I,I)
      TYPE 47,IR
   47 FORMAT(10X,/RESIDUE =,1220.10,/)
      TYPE 14
   14 FORMAT(10X,14HG-MATRIX BELOW,/)
      DO 15 I=1,NP
      DO 15 J=1,NP
      V4R(I,J)=REAL(G(I,J))
   15 V4I(I,J)=AIMAG(G(I,J))
      TYPE 5
      TYPE 13,((V4R(I,J),J=1,NP),I=1,NP)
      TYPE 10
      TYPE 13,((V4I(I,J),J=1,NP),I=1,NP)
      TYPE 2
      TYPE 16
   16 FORMAT(10X,14HB-MATRIX BELOW,/)
      DO 17 I=1,NP
      DO 17 J=1,NP
      V4R(I,J)=REAL(B(I,J))
   17 V4I(I,J)=AIMAG(B(I,J))
      TYPE 6
      TYPE 13,((V4R(I,J),J=1,NP),I=1,NP)
      TYPE 10
      TYPE 13,((V4I(I,J),J=1,NP),I=1,NP)
      TYPE 7
      IR=0.
      DO 46 I=1,NP
   46 IR=IR+V4R(I,1)
      TYPE 17,IR
      TYPE 18
   18 FORMAT(1X,26HFORWARD POWER MATRIX BELOW,/)
```

```
   DO 19 I=1,NP
   DO 19 J=1,NP
   V4R(I,J)=REAL(F(I,J))
19 V4I(I,J)=AIMAG(F(I,J))
   TYPE 6
   TYPE 13,((V4R(I,J),J=1,NP),I=1,NP)
   TYPE 10
   TYPE 13,((V4I(I,J),J=1,NP),I=1,NP)
   TYPE 9
   TYPE 20
20 FORMAT(1X,27HBACKWARD POWER MATRIX BELOW,/)
   DO 21 I=1,NP
   DO 21 J=1,NP
   V4R(I,J)=REAL(FP(I,J))
21 V4I(I,J)=AIMAG(FP(I,J))
   TYPE 6
   TYPE 13,((V4R(I,J),J=1,NP),I=1,NP)
   TYPE 10
   TYPE 13,((V4I(I,J),J=1,NP),I=1,NP)
   TYPE 9
   TYPE 22
22 FORMAT(1X,43HFORWARD REFLECTION COEFFICIENT MATRIX BELOW,/)
   DO 23 I=1,NP
   DO 23 J=1,NP
   V4R(I,J)=REAL(CN(I,J))
23 V4I(I,J)=AIMAG(CN(I,J))
   TYPE 6
   TYPE 13,((V4R(I,J),J=1,NP),I=1,NP)
   TYPE 10
   TYPE 13,((V4I(I,J),J=1,NP),I=1,NP)
   TYPE 9
   TYPE 24
24 FORMAT(1X,44HBACKWARD REFLECTION COEFFICIENT MATRIX BELOW,/)
   DO 25 I=1,NP
   DO 25 J=1,NP
   V4R(I,J)=REAL(CNP(I,J))
25 V4I(I,J)=AIMAG(CNP(I,J))
   TYPE 6
   TYPE 13,((V4R(I,J),J=1,NP),I=1,NP)
   TYPE 10
   TYPE 13,((V4I(I,J),J=1,NP),I=1,NP)
   TYPE 9
   DO 90 I=1,NP
   DO 90 J=1,NP
90 AEIG(I,J)=F(I,J)
   CALL VCVTCH(AEIG,NP,4,AF(6))
   CALL EIGCH(AEIG,NP,0,EL,V1,4,WK,IER)
   TYPE 29
29 FORMAT(1X,20HEIGENVALUES OF E BELOW,/)
   TYPE 13,(EL(I),I=1,NP)
   TYPE 9
   SUM=0.
   DO 60 I=1,NP
60 SUM=SUM+EL(I)
   TYPE 30,SUM
30 FORMAT(1X,13HSUM = TRACE =,1E20.10,//)
   TYPE 9
   DO 91 I=1,NP
   DO 91 J=1,NP
91 AEIG(I,J)=B(I,J)
```

```
      CALL VCVTCH(AEIG,NP,4,AEIG)
      CALL EIGCH(AEIG,NP,0,EL,VI,4,WK,IER)
      SUM=0.
      DO 51 I=1,NP
   51 SUM=SUM+EL(I)
      TYPE 39
   39 FORMAT(1X,22HEIGENVALUES OF B BELOW,/)
      TYPE 13,(EL(I),I=1,NP)
      TYPE 9
      TYPE 36,SUM
   80 CONTINUE
      DO 92 I=1,NP
      DO 92 J=1,NP
   92 AEIG(I,J)=P(I,J)
      CALL VCVTCH(AEIG,NP,4,AEIG)
      CALL EIGCH(AEIG,NP,0,EL,VI,4,WK,IER)
      IF(EL(1).LT.0.)GO TO 94
      GO TO 95
   94 TYPE 96
   96 FORMAT(1X,48HMATRIX NOT POSITIVE DEFINITE--PROGRAM TERMINATED,/)
      TYPE 97,EL(1)
   97 FORMAT(1X,21HSMALLEST EIGENVALUE =,1E20.10)
      STOP
   95 CONTINUE
        SUM=0.
        DETP(IFP1)=1.
        ELP(IFP1,1)=EL(NP)
        ELP(IFP1,2)=EL(1)
      DO 62 I=1,NP
      DETP(IFP1)=DETP(IFP1)*EL(I)
   62 SUM=SUM+EL(I)
      FACT=(ELNUM+1.+ELNUM)/((FCD...-1.)-ENUM)
      FPE(IFP1)=DETP(IFP1)*(FACT**NP)
      IF(19.LT.NFRS)GO TO 81
      TYPE 33
   33 FORMAT(1X,21HEIGENVALUES OF P BELOW,/)
      TYPE 13,(EL(I),I=1,NP)
      TYPE 36,SUM
      TYPE 9
   81 CONTINUE
      DO 93 I=1,NP
      DO 93 J=1,NP
   93 AEIG(I,J)=PP(I,J)
      CALL VCVTCH(AEIG,NP,4,AEIG)
      CALL EIGCH(AEIG,NP,0,EL,VI,4,WK,IER)
      IF(EL(1).LT.0.)GO TO 104
      GO TO 105
  104 TYPE 96
      TYPE 97,EL(1)
      STOP
  105 CONTINUE
        SUM=0.
        DETPP(IFP1)=1.
        ELPP(IFP1,1)=EL(NP)
      ELPP(IFP1,2)=EL(1)
      DO 63 I=1,NP
      DETPP(IFP1)=DETPP(IFP1)*EL(I)
   63 SUM=SUM+EL(I)
      IF(19.LT.NPRS)GO TO 82
      TYPE 35
   35 FORMAT(1X,26HEIGENVALUES OF P PRIME BELOW,/)
```

```
        TYPE 13,(CC(I),I=1,NP)
       TYPE 80,SQ
       TYPE 9
       TYPE 39
   39 FORMAT(1X,33HFORWARD FILTER COEFFICIENTS BELOW,/)
       DO 70 K=1,IFP1
       DO 71 I=1,NP
       DO 71 J=1,NP
       V4R(I,J)=REAL(CF(I,J,K))
   71 V4I(I,J)=AIMAG(CF(I,J,K))
       TYPE 6
       TYPE 13,((V4R(I,J),J=1,NP),I=1,NP)
       TYPE 10
       TYPE 13,((V4I(I,J),J=1,NP),I=1,NP)
   70 CONTINUE
       TYPE 73
   73 FORMAT(1X,34HBACKWARD FILTER COEFFICIENTS BELOW,/)
       DO 74 K=1,IFP1
       DO 75 I=1,NP
       DO 75 J=1,NP
       V4R(I,J)=REAL(CB(I,J,K))
   75 V4I(I,J)=AIMAG(CB(I,J,K))
       TYPE 6
       TYPE 13,((V4R(I,J),J=1,NP),I=1,NP)
       TYPE 10
       TYPE 13,((V4I(I,J),J=1,NP),I=1,NP)
   74 CONTINUE
       TYPE 9
   62 CONTINUE
       RETURN
       END
C*****************************************************************
        SUBROUTINE UPDAT
        COMMON/COMUN/K1,I1,CF,CB,C,CP,E,B,G,SS,CN,CNP,BI,EI,BBB,IP,CNDAT,WN
      1,NN,N,NI,IFP,P,PP,NMI
        COMMON/FLIP/AF16,UK,EINV
        COMPLEX U5,U6,AF16,EINV
        COMPLEX BI,FI,CB,CF,C,CI,...,E,G,SS,CN,CNP,BI,EI,BBB,V1,V2,V3,CNDAT
      1,UN,CNAI,CI,P,PP,PII,PPI,U1,U2,U11,II,U,AA,BB,CC,V4,V5
        DIMENSION BI(4, 300),EI(4, 300),CB(4,4,40),CF(4,4,40),C(4,4),CP(4
      1,4),E(4,4),B(4,4),G(4,4),SS(4,4),CN(4,4),CNP(4,4),BI(4,4),EI(4,4),
      2BBB(16),V1(4,4),V2(4,4),V3(4,4),P(4,4),PP(4,4),PII(4,4),PPI(4,4)
        DIMENSION V4(4,4),V5(4,4),V6(4,4),V7(4,4),U1(4,4),U2(4,4),U11(4,4
      1),II(4,4),U(4,4),AA(4,4),BB(4,4),CC(4,4),AF16(4,4),EINV(16,16)
        DIMENSION U5(4,4,40),U6(4,4,40),UK(12)
          AI=FLOAT(I9)
          CI=CMPLX(AI,0.)
          CNAI=CN
          NP=NCP
          NNI=NNAI-1
          ENDAI=FLOAT(NDAI)
          CNAI=CMPLX(ENDAI,0.)
          CNAI=CNDAI-CI
          WN=CMPLX(1.,0.)/CNAI
          IFP1=IFP+1
          NNMI=NN-I+1
       DO 9 K=1,NNI
       KPI=K+I
       DO 9 I=1,NP
     9 FI(I,K)=EI(I,KPI)
```

```
      DO 1 I=1,NP
      DO 1 J=1,NP
        E(I,J)=CMPLX(0.,0.)
        F(I,J)=CMPLX(0.,0.)
        G(I,J)=CMPLX(0.,0.)
       DO 2 K=1,NQ
      E(I,J)=E(I,J)+[illegible](I,K)*CONJG(B(J,K))
      F(I,J)=F(I,J)+B(I,K)*[illegible](J,K))
    2 G(I,J)=G(I,J)+FI(I,K)*CONJG(EI(J,K))
        E(I,J)=E(I,J)*WA
        F(I,J)=F(I,J)*WA
        G(I,J)=G(I,J)*WB
        FI(I,J)=F(I,J)
        FF(I,J)=FP(I,J)
      B1(I,J)=G(I,J)
    1 E1(I,J)=E(I,J)
      DO 30 I=1,NP
      DO 30 J=1,NP
   30 EINV(I,J)=B1(I,J)
      CALL CMINV(EINV,NP,BBB)
      DO 31 I=1,NP
      DO 31 J=1,NP
   31 B1(I,J)=EINV(I,J)
      DO 32 I=1,NP
      DO 32 J=1,NP
   32 EINV(I,J)=E1(I,J)
      CALL CMINV(EINV,NP,BBB)
      DO 33 I=1,NP
      DO 33 J=1,NP
   33 E1(I,J)=EINV(I,J)
      DO 34 I=1,NP
      DO 34 J=1,NP
   34 EINV(I,J)=FI(I,J)
      CALL CMINV(EINV,NP,BBB)
      DO 35 I=1,NP
      DO 35 J=1,NP
   35 FI(I,J)=EINV(I,J)
      DO 36 I=1,NP
      DO 36 J=1,NP
   36 EINV(I,J)=FF(I,J)
      CALL CMINV(EINV,NP,BBB)
      DO 37 I=1,NP
      DO 37 J=1,NP
   37 FF(I,J)=EINV(I,J)
C     STEPS BETWEEN DASHED LINES DEFINE WEIGHT MATRICES Q1 AND Q2 AND
C      ALSO Q1 INVERSE
C     - - - - - - -   - - - - - - - - - - - - - - - - - - - - - -
      DO 10 I=1,NP
      DO 10 J=1,NP
      Q1(I,J)=FI(I,J)
      Q1I(I,J)=F(I,J)
   10 Q2(I,J)=FFI(I,J)
C      ABOVE STEPS GIVE INVERSE POWER WEIGHTING.
C     - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
      DO 11 I=1,NP
      DO 11 J=1,NP
      V1(I,J)=CMPLX(0.,0.)
      DO 11 K=1,NP
   11 V1(I,J)=V1(I,J)+PP(I,K)*Q2(K,J)
      DO 12 I=1,NP
      DO 12 J=1,NP
      T(I,J)=CMPLX(0.,0.)
```

```
      DO 12 K=1,NP
   12 II(I,J)=II(I,J)+V1(I,K)*FF(K,J)
      DO 38 I=1,NP
      DO 38 J=1,NP
   38 EIGV(I,J)=II(I,J)
      CALL CHIGV(EIGV,NP,NP,NSF)
      DO 39 I=1,NP
      DO 39 J=1,NP
   39 V1(I,J)=EIGV(I,J)
      DO 13 I=1,NP
      DO 13 J=1,NP
      V2(I,J)=CMPLX(0.,0.)
        V3(I,J)=CMPLX(0.,0.)
      DO 13 K=1,NP
      V2(I,J)=V2(I,J)+FII(I,K)*E(K,J)
   13 V3(I,J)=V3(I,J)+FII(I,K)*QII(K,J)
      DO 14 I=1,NP
      DO 14 J=1,NP
        G(I,J)=CMPLX(0.,0.)
        V5(I,J)=CMPLX(0.,0.)
      DO 14 K=1,NP
      G(I,J)=G(I,J)+V2(I,K)*FII(K,J)
   14 V5(I,J)=V5(I,J)+G(I,K)*V3(K,J)
      DO 15 I=1,NP
      DO 15 J=1,NP
      AA(I,J)=CMPLX(0.,0.)
        BB(I,J)=CMPLX(0.,0.)
        CC(I,J)=CMPLX(0.,0.)
      DO 15 K=1,NP
      AA(I,J)=AA(I,J)+II(I,K)*E(K,J)
      BB(I,J)=BB(I,J)+G(I,K)*QII(K,J)
   15 CC(I,J)=CC(I,J)-II(I,K)*G(K,J)-FF(I,K)*V5(K,J)
      CALL QSTCN(AA,BB,CC,NP,CN)
      DO 25 I=1,NP
      DO 25 J=1,NP
      V1(I,J)=CMPLX(0.,0.)
      DO 25 K=1,NP
   25 V1(I,J)=V1(I,J)+CONJG(CN(K,I))*FF(K,J)
      DO 26 I=1,NP
      DO 26 J=1,NP
      CNP(I,J)=CMPLX(0.,0.)
      DO 26 K=1,NP
   26 CNP(I,J)=CNP(I,J)+FII(I,K)*V1(K,J)
      DO 27 I=1,NP
      DO 27 J=1,NP
      DO 27 K=1,IFP1
      U6(I,J,K)=C(I,J,K)
   27 U5(I,J,K)=CI(I,J,K)
      DO 5 K=1,IFP1
      IP2MK=IFP+2-K
      DO 4 I=1,NP
      DO 4 J=1,NP
      DO 4 KK=1,NP
      CF(I,J,K)=CF(I,J,K)+U6(I,KK,IP2MK)*CN(KK,J)
    4 CB(I,J,K)=CB(I,J,K)+U5(I,KK,IP2MK)*CNP(KK,J)
      DO 5 K=1,NAI
      DO 9 I=1,NP
      V3(I,1)=AI(I,K)
    9 V3(I,2)=BI(I,K)
```

```
      DO 5 K=1,NP
      DO 5 L=1,NP
      P(I,K)=P(I,K)+CONJG(CN(K,I))*Q1(K,J)
    5 Q(I,K)=P(I,K)+CONJG(CNP(K,I))*Q2(K,J)
      DO 6 I=1,NP
      DO 6 J=1,NP
      Q1(I,J)=CMPLX(0.,0.)
        Q2(I,J)=CMPLX(0.,0.)
      DO 6 K=1,NP
      Q1(I,J)=Q1(I,J)+PP(I,K)*CN(K,J)
    6 Q2(I,J)=Q2(I,J)+P(I,K)*CNP(K,J)
      DO 7 I=1,NP
      DO 7 J=1,NP
      DO 7 K=1,NP
      P(I,J)=P(I,J)-CONJG(CN(K,I))*Q1(K,J)
    7 PP(I,J)=PP(I,J)-CONJG(CNP(K,I))*Q2(K,J)
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE GETCN(A,B,C,N,CN)
      DIMENSION A(4,4),B(4,4),C(4,4),CN(4,4),PP(16),AB(16,16)
      COMPLEX A,B,C,BP,CN,AB
C  THIS SUBROUTINE SOLVES THE MATRIX EQUATION A*(CN)+(CN)*B=C FOR THE
C   REFLECTION COEFFICIENT,CN.    THE METHOD IS THAT OF BELLMAN,MATRIX
C   ANALYSIS,OLD EDITION,PAGE 231.
      N2=N*N
      DO 1 I=1,N
      DO 1 J=1,N
      II=(I-1)*N+J
    1 BP(II)=C(I,J)
      CALL KRONECK(A,B,AB,N)
      CALL [illegible](AB,N2,BP)
      DO 2 I=1,N
      DO 2 J=1,N
      II=(I-1)*N+J
    2 CN(I,J)=BP(II)
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE KRONECK(A,B,AB,N)
      DIMENSION A(4,4),B(4,4),AB(16,16)
      COMPLEX A,B,AB
C  THIS SUBROUTINE FORMS THE(N**2)X(N**2) MATRIX AB WHICH HAS THE
C  FORM AB=(AXI)+(IXB) WHERE THE CROSSES DENOTE KRONECKER PRODUCTS.
C  SEE BELLMAN,MATRIX ANALYSIS,OLD EDITION,P. 231.
      N2=N*N
      DO 1 I=1,N2
      DO 1 J=1,N2
    1 AB(I,J)=CMPLX(0.,0.)
      DO 2 I=1,N
      DO 2 J=1,N
      DO 2 K=1,N
      I1=(I-1)*N+K
        I2=(J-1)*N+K
    2 AB(I1,I2)=A(I,J)
      DO 3 I=1,N
      DO 3 J=1,N
      DO 3 K=1,N
      I1=(I-1)*N+J
        I2=(I-1)*N+K
    3 AB(I1,I2)=AB(I1,I2)+B(K,J)
      RETURN
      END
```

```
C ------------------------------------------------------------------
      [illegible] (N,A,B)
      [illegible] (16),A(16,16),B(16),INDEX(16,2),IPIVOT(16)
      [illegible] AMAX,T,SWAP,B,PIVOT
      DO [illegible] I=1,N
   20 IPIVOT(I)=0
      DO 550 I=1,N
      AMAX=CMPLX(0.,0.)
      DO 105 J=1,N
      IF(IPIVOT(J).EQ.1)GO TO 101
      DO 100 K=1,N
      IF(IPIVOT(K)-1)80,100,740
   80 A1=CABS(AMAX)
      A12=CABS(A(J,K))
      IF(A1[illegible])85,100
   85 IROW=J
      ICOLUM=K
      AMAX=A(J,K)
  100 CONTINUE
  101 CONTINUE
      IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
      IF(IROW.EQ.ICOLUM)GO TO 260
      DO 200 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
      SWAP=B(IROW)
      B(IROW)=B(ICOLUM)
      B(ICOLUM)=SWAP
  260 INDEX(I,1)=IROW
      INDEX(I,2)=ICOLUM
      PIVOT(I)=A(ICOLUM,ICOLUM)
      A(ICOLUM,ICOLUM)=CMPLX(1.,0.)
      DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)
      B(ICOLUM)=B(ICOLUM)/PIVOT(I)

      DO 550 L1=1,N
      IF(L1.EQ.ICOLUM)GO TO 550
  400 T=A(L1,ICOLUM)
      A(L1,ICOLUM)=CMPLX(0.,0.)
      DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
      B(L1)=B(L1)-B(ICOLUM)*T
  550 CONTINUE
      DO 710 I=1,N
      L=N+1-I
      IF(INDEX(L,1).EQ.INDEX(L,2))GO TO 710
      JROW=INDEX(L,1)
      JCOLUM=INDEX(L,2)
      DO 705 K=1,N
      SWAP=A(K,JROW)
      A(K,JROW)=A(K,JCOLUM)
      A(K,JCOLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
  740 RETURN
      END
C ------------------------------------------------------------------
      SUBROUTINE FFT
      COMMON FFU[illegible]/A(256),N,M,[illegible]
      COMPLEX A,[illegible]
```

```
      [illegible]
      [illegible]
      [illegible]
      CALL [illegible]
      [illegible]
      DO [illegible] = 1,N
         [illegible]
         [illegible]
         U = CMPLX(1.,0.)
         DO 60 J = 1,LE1
            DO 20 I = J,N,LE
               IP = I+LE1
               T = A(IP)*U
               A(IP) = A(I) - T
               A(I) = A(I) + T
 20         CONTINUE
            U = U*W(L)
 60      CONTINUE
 70   CONTINUE
      RETURN
      END
C ------------------------------------------------------------------------
      SUBROUTINE [illegible]
      COMMON/[illegible]/A(256),M,N,W(15)
      COMMON/[illegible]
      COMPLEX A,U,W,T
      PI=3.141592654
      [illegible] N
      [illegible]
      DO 10 I = 1,M
         X = COS(PI/[illegible])
         Y = [illegible](PI/[illegible])
         [illegible]
         [illegible]
 10   CONTINUE
      RETURN
      END
C ------------------------------------------------------------------------
      SUBROUTINE [illegible]
C        COMMON/[illegible]/A(256),M,N,W(10)
      COMPLEX A,S,T
      N2 = N/2
      NM1 = N-1
      J = 1
      DO 3 I=1,NM1
C           IF(I.GE.J) GO TO 1
            T = A(J)
            A(J) = A(I)
C              A(I) = T
C 1         CONTINUE
         K = N2
C        CONTINUE
            IF(K.GE.J) GO TO 25
            J = J - K
C              K = K/2
C           GO TO 2
 25      CONTINUE
         J = J + K
 3    CONTINUE
      RETURN
      END
```

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A111 433	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Further Study of the Multichannel Maximum Entropy Spectral Analysis	5. TYPE OF REPORT & PERIOD COVERED Technical Report
	6. PERFORMING ORG. REPORT NUMBER SMU-EE-TR-82-2
7. AUTHOR(s) C. H. Chen and Gia-Kinn Young	8. CONTRACT OR GRANT NUMBER(s) N00014-79-C-0494
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Electrical Engineering Southeastern Massachusetts University North Dartmouth, Massachusetts 02747	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 042-422
11. CONTROLLING OFFICE NAME AND ADDRESS Statistics and Probability Program Office of Naval Research, Code 436 Arlington, Virginia 22217	12. REPORT DATE February 1, 1982
	13. NUMBER OF PAGES 41
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Maximum entropy power spectrum; multichannel spectral analysis; cross power spectrum; estimation of phase difference; coherence; eigenspectrum analysis; line-splitting and frequency shifting; computer program listing.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Based on the multichannel generalization of the Burg's maximum entropy spectral analysis, the report reviews the computing procedures and closely examines the software realization problem by using sinusoids as examples. The high-resolution auto- and cross spectra available are practically significant. For example, the phase difference estimate is useful for time delay estimation. The line-splitting and frequency shifting phenomena are more evident in real signals than in multichannel complex signals. Detailed computer program listing is provided.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)


FILMED
8